Advant® OCS
with Master software

Engineering Methods
Advant® Station 500 Series
Engineering Station

User’s Guide

ABB

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaCommand, AdvaInform, AdvaBuild, and AdvaControl are registered trademarks of ABB Asea Brown Boveri Ltd. Switzerland

AutoCAD and DXF are registered trademarks of Autodesk Inc.

FrameMaker is a trademark of Frame Technology Corp.

HP and all HP products described in this document are registered trademarks of Hewlett Packard Company

IBM-PC is a trademark of IBM Corporation. (International Buisiness Machines Corp.)

MOD 300 is a trademark of ABB Process Automation Inc.

ORACLE and SQL*ReportWriter are registered trademarks of Oracle Corporation

OSF/Motif is a trademark of The Open Software Foundation, Inc.

UNIX is a trademark of AT&T Corp.

X-Windows Systems are trademarks of Microsoft

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems AB . ABB Industrial Systems AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems AB ’s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

CE MARKING

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 72/23/EEC.

Copyright © ABB Industrial Systems AB 1997

3BSE001264/D

Template: 3BSE001286/E

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 General Information 1-1
1.2 Manual Organization 1-2
1.3 Related Documentation 1-2

Chapter 2 - AdvaBuild Overview

2.1 AdvaBuild Environment 2-1
2.2 AdvaBuild Builders 2-1
    2.2.1 Structure Builder 2-1
    2.2.2 Template Builder 2-1
    2.2.3 Function Chart Builder 2-2
    2.2.4 Plant Documentation Builder 2-2
    2.2.5 Diagram Builder 2-2
    2.2.6 Electrical Diagram Builder 2-2
    2.2.7 Document Builder 2-2
    2.2.8 Display Builder 2-2
    2.2.9 On-Line Builder 2-2

Chapter 3 - Principles of Project Engineering with AdvaBuild

3.1 Scope of Supported Engineering Activities 3-1
3.2 Object-oriented Engineering 3-2
3.3 Structuring of Plants 3-2
    3.3.1 General Structuring Principles 3-2
    3.3.2 Plant/Function Oriented Structure 3-3
    3.3.3 Location Oriented Structure 3-4
    3.3.4 Control System Oriented Structure 3-5
    3.3.5 Plant Documentation Structure 3-6
    3.3.6 Relations between CAPE-objects 3-7
        3.3.6.1 Links Between Objects 3-7
        3.3.6.2 Solutions (Engineering Units) 3-8

Chapter 4 - General Engineering Guidelines

4.1 Creation of the Plant Structures 4-1
    4.1.1 Plant/Function Structure ("="-structure) 4-1
        4.1.1.1 Object-types in the Function Structure 4-2
        4.1.1.2 Link-types to Objects in the Function Structure 4-4
        4.1.1.3 Designation of Objects in the Function Structure 4-5
        4.1.1.4 Preconditions for the Design of the Function Structure ... 4-6

CONTENTS (continued)

- 4.1.2 Topographic-location Structure ("++"-structure) 4-7
  - 4.1.2.1 Object-types in the Topographic-location Structure 4-8
  - 4.1.2.2 Link-types to Objects in the Topographic-Location Structure 4-9
  - 4.1.2.3 Designation of Objects in the Topographic-location Structure 4-9
  - 4.1.2.4 Preconditions for the Design of the Topographic-location Structure 4-10
- 4.1.3 Mounting-location Structure ("+"-structure) 4-11
  - 4.1.3.1 Object-types in the Mounting-location Structure........... 4-12
  - 4.1.3.2 Link-types to Objects in the Mounting-location Structure 4-13
  - 4.1.3.3 Designation of Objects in the Mounting-location Structure 4-14
  - 4.1.3.4 Preconditions for the Design of the Mounting-location Structure 4-14
- 4.1.4 Control System Structure ("#"-structure)... 4-15
  - 4.1.4.1 Object-types in the Control System Structure 4-16
  - 4.1.4.2 Special Hints for the Creation of the Control System Structure... 4-17
  - 4.1.4.3 Link-types to Objects in the Control System Structure.. 4-19
  - 4.1.4.4 Designation of Objects in the Control System Structure 4-19
  - 4.1.4.5 Preconditions for the Design of the Control System Structure... 4-20
- 4.1.5 Plant Documentation Structure ("&"-structure)................................... 4-21
  - 4.1.5.1 Object-types in the Plant Documentation Structure 4-22
  - 4.1.5.2 Link-types to Objects in the Plant Documentation Structure 4-22
  - 4.1.5.3 Designation of Objects in the Plant Documentation Structure 4-23

4.2 List and Tree Presentation of Objects.. 4-23

4.3 Creating Text, Drawing, and Diagram Objects.. 4-25
- 4.3.1 Creating and Editing TEXT-objects.. 4-26

4.4 Importing Text and Drawing/diagram Objects 4-27
- 4.4.1 Importing TEXT-objects... 4-27
- 4.4.2 Importing Drawing/diagram Objects 4-28

4.5 Solutions.. 4-29
- 4.5.1 Definition of Solutions... 4-29
- 4.5.2 Content of Solutions... 4-29

4.6 Multi-user Engineering 4-33

CONTENTS (continued)

Chapter 5 - Basic Engineering

5.1 Opening a Project.......... 5-1
5.2 Collecting General Project Information.......... 5-2
5.3 Design of the Basic Plant Structures.......... 5-3
    5.3.1 Design of the Function Structure.......... 5-3
    5.3.2 Design of the Topographic-location Structure.......... 5-3
5.4 Creation/Selection of Project Standards.......... 5-3
    5.4.1 Type Circuits.......... 5-3
    5.4.2 Item Designation.......... 5-4
    5.4.3 Man Machine Communication.......... 5-4
    5.4.4 DCS Software Standards.......... 5-4
    5.4.5 DCS Hardware Standards.......... 5-4
    5.4.6 Test Procedures.......... 5-5
    5.4.7 Installation Standards.......... 5-5
    5.4.8 Import/Export Standards.......... 5-5
    5.4.9 Design Standards.......... 5-5
5.5 Basic Function Planning and Acquisition of Basic Engineering-data.......... 5-6
    5.5.1 Handling of P&I-diagrams.......... 5-6
    5.5.2 Assignment of Objects in the Topographic-location Structure to DangerAreas of the Plant.......... 5-9
    5.5.3 Specification of Man Machine Communication.......... 5-9
    5.5.4 Acquisition of Data for Measuring, Actuating, and Consuming Functions.......... 5-9
    5.5.5 Definition of Hierarchical Relations of Control Functions.......... 5-13
    5.5.6 Basic Specification of Control Functions and Assignment to Plant Structure Objects.......... 5-14
5.6 Basic Configuration of the Control System.......... 5-16
    5.6.1 Design of the Control System Structure.......... 5-16
5.7 Basic Design of the Arrangement and Interconnection.......... 5-17
    5.7.1 Design of the Mounting-location Structure.......... 5-17
    5.7.2 Acquisition of Basic Information for Arrangement and Interconnection.......... 5-18
        5.7.2.1 Basic Arrangement Planning.......... 5-18
        5.7.2.2 Specifications of Interconnections between Constructional Units (Objects of the Mounting-location Structure) and to Single Components.......... 5-18
5.8 Planning of Documentation.......... 5-20
    5.8.1 Planning of the Document-types within a Project.......... 5-20
    5.8.2 Creation of the Document Structure.......... 5-23

CONTENTS (continued)

Chapter 6 - Detailed Engineering

6.1 Detailed Tag Engineering 6-1
- 6.1.1 Objects and Information in the Substructure of Tags 6-1
  - 6.1.1.1 DCS-signal Elements 6-1
  - 6.1.1.2 Device Function Objects and Device Objects 6-3
  - 6.1.1.3 Document Objects 6-6

6.2 Detailed Configuration of the Control System........ 6-7

6.3 Detailed Design of Arrangement and Interconnection........ 6-9
- 6.3.1 Design of Constructional Units and Device Arrangement........ 6-9
- 6.3.2 Arrangement of Constructional Units and Components 6-12
- 6.3.3 Detailed Interconnection Engineering........ 6-13

6.4 Design of Mechanical Assemblies (Hook-ups)........ 6-14

6.5 Detail Design of Control Functions 6-15
- 6.5.1 Handling of Type Circuits........ 6-15
- 6.5.2 Assignment of Control Functions to Control System Objects (Nodes) 6-16
- 6.5.3 Configuration of Control Functions 6-16
- 6.5.4 Insertion of MasterPiece Data base Elements........ 6-19
- 6.5.5 Linking a DCS-signal Object to an Open Channel of a Board-object .. 6-20
- 6.5.6 Loading of PC-programs and DB-elements to the Target System........ 6-22

6.6 Creating Displays 6-25

6.7 Plant Documentation........ 6-28

Appendix A - Overview of Object-types and Link-types

Appendix B - Target Systems Supported by AdvaBuild

INDEX

Chapter 1 Introduction

1.1 General Information

This manual describes methods for plant engineering using the following AdvaBuild software packages:

- AdvaBuild Basic Functions
  - structure Builder
  - template Builder
  - function Chart Builder
  - plant Documentation Builder.
- AdvaBuild Optional Software
  - document Builder
  - diagram Builder
  - electrical Diagram Builder
  - On-Line Builder
  - display Builder.

Collectively, these software packages realize a Computer Aided Plant Engineering (CAPE) concept that integrates plant engineering CAD capabilities with tools for DCS configuration. Together, these software packages support the following project engineering functions:

- configuration of the ABB MASTER DCS-system
- configuration of field instrumentation and field equipment
- project documentation.

The AdvaBuild software packages support an object-oriented method of system engineering. This book describes the principles of object-oriented engineering, and provides guidelines for object-oriented engineering with AdvaBuild.

The AdvaBuild software runs on the Advant Station 500 Series Engineering Stations.
The Engineering Stations can be used as stand-alone stations with a single workplace, or they can be used with multiple workplaces by adding X-terminals, thus providing a multi-user environment for engineering.

This book is intended as a guide for plant engineering using AdvaBuild. Instructions for operating the various builders are provided in other instruction books as described in Section 1.3, Related Documentation.

1.2 Manual Organization

This manual is organized into the following chapters:

- Chapter 1, Introduction
- Chapter 2, AdvaBuild Overview
- Chapter 3, Principles of Project Engineering with AdvaBuild
- Chapter 4, General Engineering Guidelines
- Chapter 5, Basic Engineering
- Chapter 6, Detailed Engineering
- Appendix A, Overview of Object-types and Link-types
- Appendix B, Target Systems Supported by AdvaBuild.

1.3 Related Documentation

A complete listing of AdvaBuild Documentation is shown in Figure 1-1 and is described below:

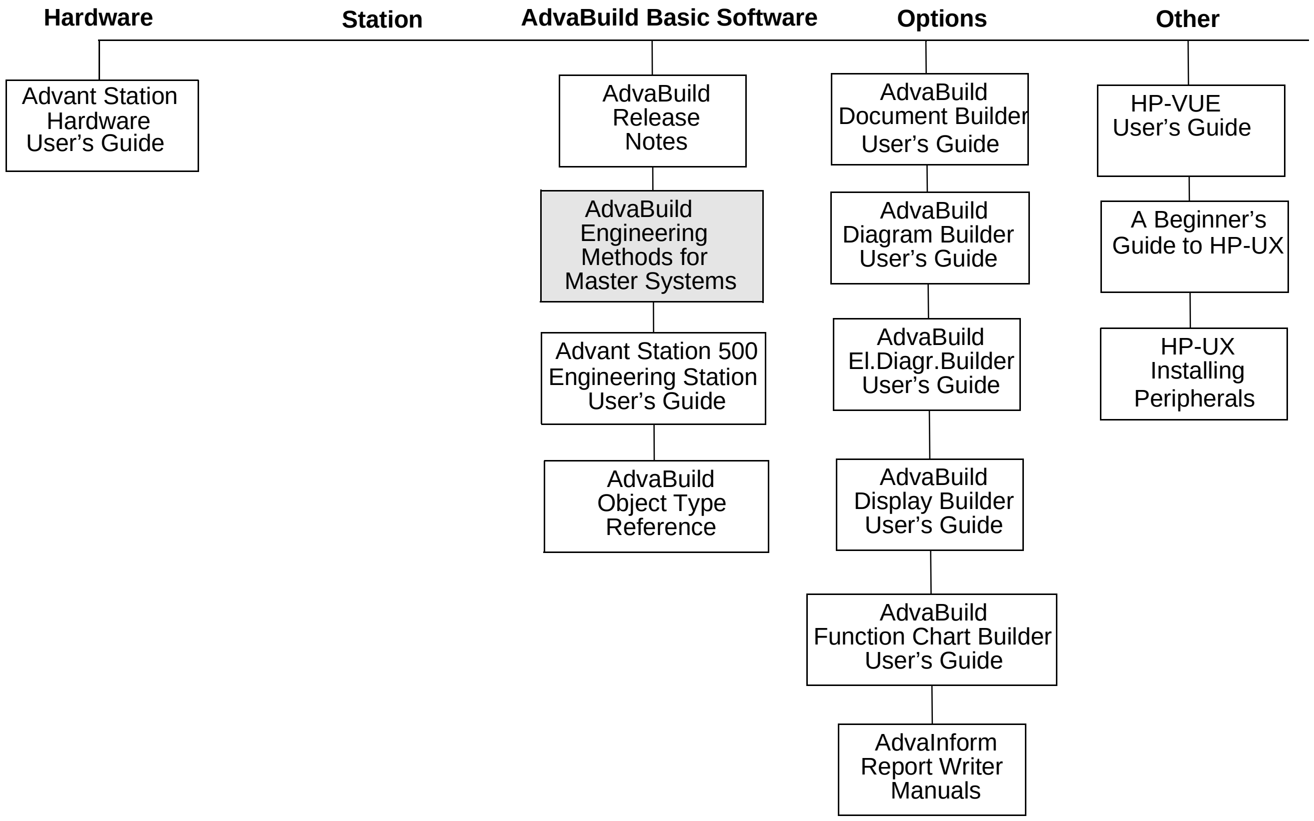

Hardware
Station
AdvaBuild Basic Software
Options
Other
Advant Station Hardware User's Guide
AdvaBuild Release Notes
AdvaBuild Engineering Methods for Master Systems
Advant Station 500 Engineering Station User's Guide
AdvaBuild Object Type Reference
AdvaBuild Document Builder User's Guide
AdvaBuild Diagram Builder User's Guide
AdvaBuild El.Diagr.Builder User's Guide
AdvaBuild Display Builder User's Guide
AdvaBuild Function Chart Builder User's Guide
AdvaInform Report Writer Manuals
HP-VUE User's Guide
A Beginner's Guide to HP-UX
HP-UX Installing Peripherals

Figure 1-1. Related Documentation

- *Advant Station 515 Hardware User's Guide* or
  Advant Station 520i Hardware User's Guide

  Consult this book for instructions regarding hardware setup and installation.

- Release Notes

  Release notes are included in the delivery file with AdvaBuild software. The release notes provide installation instructions for both initial and upgrade installations, information regarding changes that have occurred since the last release of the project, and so on.

- *Engineering Methods Advant Station 500 Series Engineering Systems*

  This book describes the principles of object-oriented engineering with AdvaBuild.

- A*dvant Station 500 Engineering Station User's Guide*

  Consult this book for a general functional overview of the Engineering Station. This book also describes how to use the Structure Builder, Template Builder, Plant Documentation Builder, how to access Display Builder, Function Chart Builder and On-line Builder.

- *AdvaBuild Object Type Reference Manual*

  The Object Type Reference manual provides information on how to configure the various database object attributes.

- *AdvaBuild Document Builder User's Guide*
  This Binder is a documentation how to use the Document Builder, it includes template documentation as well.

- *AdvaBuild Diagram Builder User's Guide*
  This Binder is a description how to design drawings and diagrams with AutoCAD, including definition of user defined symbols. It includes library information as well.

- *AdvaBuild Electrical Diagram Builder User's Guide*

  This document contains a description of drawing support for function oriented electromechanical diagrams and layout drawings.

- *AdvaBuild Display Builder User's Guide*

- *AdvaBuild Function Chart Builder User's Guide*
  This book contains a description of AMPL application programs configuration.

- *AdvaInform Reports Reference Manual*
  Reference Manual for SQL*ReportWriter

- *AdvaInform Report Builder User's Guide*
  User's Guide for building Reports with SQL*ReportWriter.

Chapter 2 AdvaBuild Overview

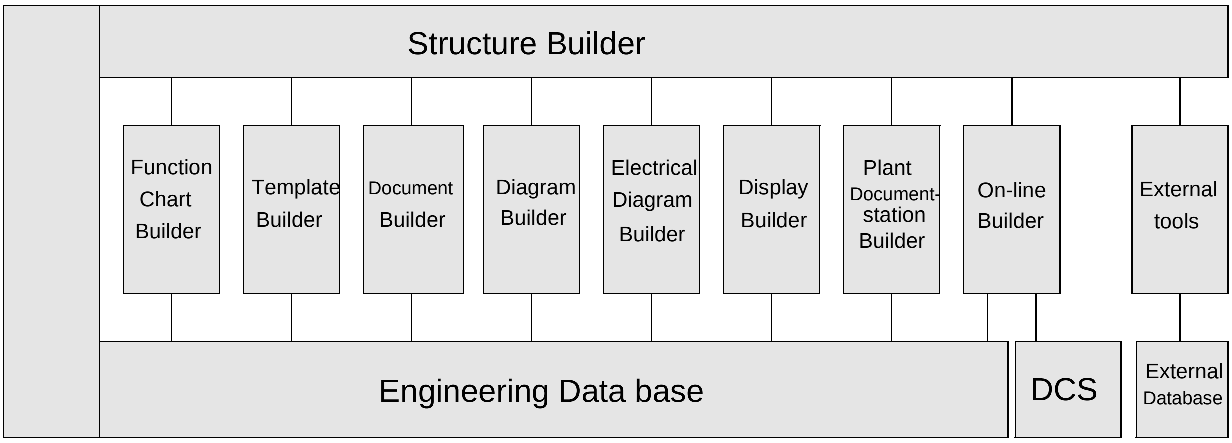


Figure 2-1. AdvaBuild Overview

2.1 AdvaBuild Environment

AdvaBuild provides a uniform environment for all Builders (integrated tools) and functions based on:

- graphical user-interface (based on X-windows, OSF/Motif)
- common project-database (based on Oracle)
- workstation (based on HP UNIX Workstation with either color- or monochrome monitor).

2.2 AdvaBuild Builders

2.2.1 Structure Builder

The Structure Builder is part of the frame system. It supports the structuring of projects in a tree-oriented, hierarchical manner, and provides the main services to users for handling data stored within the data base.

2.2.2 Template Builder

The Template Builder is designed to edit object attributes. This can be done using a single object form or a multi object form. The single object form shows the attributes for just one object at a time. The attributes may be distributed over several pages, depending upon the number of attributes. The multi object form is optimized for bulk data entry (such as process signals and IO-boards). Each row in the form represents one object. Within a multi object form, there can be an arbitrary number of objects of the same object-type.

2.2.3 Function Chart Builder

The Function Chart Builder is designed for off-line graphical configuration and documentation of AMPL (ABB Master Piece Language) application programs. When using the corresponding libraries, the Function Chart Builder can be used for:

- Advant Controller 100 Series application programs
- Advant Controller 400 Series application programs
- Master Piece 200/1 application programs.

2.2.4 Plant Documentation Builder

The Plant Documentation Builder supports the definition of documentation packages within the documentation structure and generates the required paper documentation. The definition of documentation packages is dependent on subjects of delivery and applications. The Plant Documentation Builder distinguishes up to approximately 100 different document-types which can be included in a document package. These document-types are predefined within the standard delivery of AdvaBuild but can be modified and adapted by the user.

The tool combines the document-types (as defined for the documentation packages during the configuration phase) and calls the different tools to create the corresponding documents.

2.2.5 Diagram Builder

The Diagram Builder integrates general-purpose CAD functionality into AdvaBuild. It is based on AutoCAD software. AutoCAD is a powerful drawing tool, supporting the creation of just about any engineering drawing or diagram.

2.2.6 Electrical Diagram Builder

The Electrical Diagram Builder uses the Diagram Builder as a basis. It brings additional functionality, including comfortable drawing support functions from screen menus to create and edit electrical circuit diagrams and layout drawings.

2.2.7 Document Builder

The Document Builder integrates word processing/drawing functionality into AdvaBuild. It is based on FrameMaker software. FrameMaker provides full-featured desktop publishing tools, including a complete word processor, a graphic toolbox for manipulating technical graphics, importing capabilities for text and graphics, and easy-to-use page and document design.

2.2.8 Display Builder

The Display Builder is used for the creation of graphical displays for process supervision and control via the AdvaCommand User Interface.

2.2.9 On-Line Builder

The On-Line Builder contains the communication functions between the workstation and the Master System and prepares the load and dump files.

Chapter 3 Principles of Project Engineering with AdvaBuild

3.1   Scope of Supported Engineering Activities

AdvaBuild supports project execution activities for all subjects of delivery which are handled in connection with the delivery of DCS-systems from ABB.

According to the terminology in the project model in the following figure, AdvaBuild supports the engineering activities in the following project phases:

- Basic Engineering
- Detail Engineering.

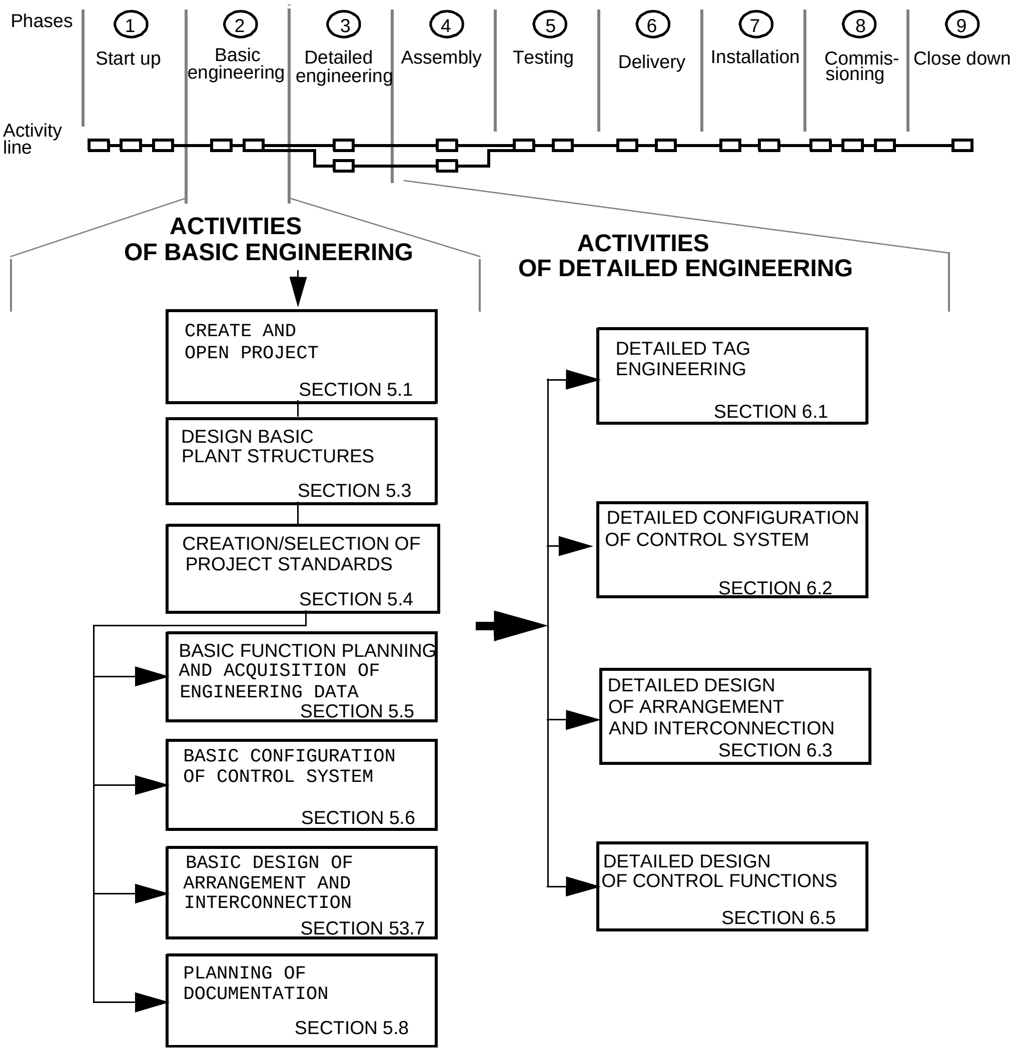

Phases
1 Start up
2 Basic engineering
3 Detailed engineering
4 Assembly
5 Testing
6 Delivery
7 Installation
8 Commis-sioning
9 Close down
Activity line
ACTIVITIES OF BASIC ENGINEERING
ACTIVITIES OF DETAILED ENGINEERING
CREATE AND OPEN PROJECT SECTION 5.1
DESIGN BASIC PLANT STRUCTURES SECTION 5.3
CREATION/SELECTION OF PROJECT STANDARDS SECTION 5.4
BASIC FUNCTION PLANNING AND ACQUISITION OF ENGINEERING DATA SECTION 5.5
BASIC CONFIGURATION OF CONTROL SYSTEM SECTION 5.6
BASIC DESIGN OF ARRANGEMENT AND INTERCONNECTION SECTION 53.7
PLANNING OF DOCUMENTATION SECTION 5.8
DETAILED TAG ENGINEERING SECTION 6.1
DETAILED CONFIGURATION OF CONTROL SYSTEM SECTION 6.2
DETAILED DESIGN OF ARRANGEMENT AND INTERCONNECTION SECTION 6.3
DETAILED DESIGN OF CONTROL FUNCTIONS SECTION 6.5

Figure 3-1. Engineering Activities

This manual provides instructions and guidelines for basic and detail engineering with AdvaBuild.

3.2 Object-oriented Engineering

The principle of engineering work with AdvaBuild is to build a data model of the plant which keeps the necessary information for erection, commissioning, operation, and maintenance. This model is constructed with CAPE-objects, which are images of the plant-objects and carry the information on plant-objects. Examples for such CAPE-objects are:

- subplant
- tag
- sensor
- motor
- DCS board
- PC-program
- cabinet.

The tools within AdvaBuild enable the user to instantiate and structure the CAPE-objects, and to handle object information. Engineering work is always related to CAPE-objects. Information may be assigned to objects at any time as far as it is available. Once data are input into the data base, the data are available for all further uses.

AdvaBuild provides an efficient set of object-types which cover plant-objects in all relevant views. An object-type is defined by an object-type name, a list of attributes, and a list of actions that can be performed on that object.

When an object-type is instantiated to represent a certain plant-object, it is positioned into a plant structure, assigned a unique object identification, and given the relevant data as attributes. Objects may be positioned into a structure view either for presentation within the tree-structure or for presentation in lists.

3.3 Structuring of Plants

3.3.1 General Structuring Principles

CAPE-objects are positioned into tree-like hierarchical plant structures. There are different structure types, each representing a certain view of the plant. AdvaBuild supports the structuring of objects according to the following views:

- plant/function oriented view
- location oriented view
- control system oriented view.

Each view is represented by a hierarchical tree-like structure of CAPE-objects. When an object is inserted into the plant model, it is inserted as child to an already defined object in one of the defined views, which are presented as tree-like structures. The topmost object, the project-object, is created when the project engineering is started. The structures are created by building up parent-child relations starting at the project object. The user navigates through these structures in order to get access to the plant objects and its data.

3.3.2 Plant/Function Oriented Structure

A plant/function oriented structure is based on the purpose of a plant. It shows the subdivision of the plant according to functions and subfunctions, without necessarily taking into account the location and/or the products implementing the functions. The objects which represent functions are hierarchically positioned in this structure as shown in Figure 3-2.

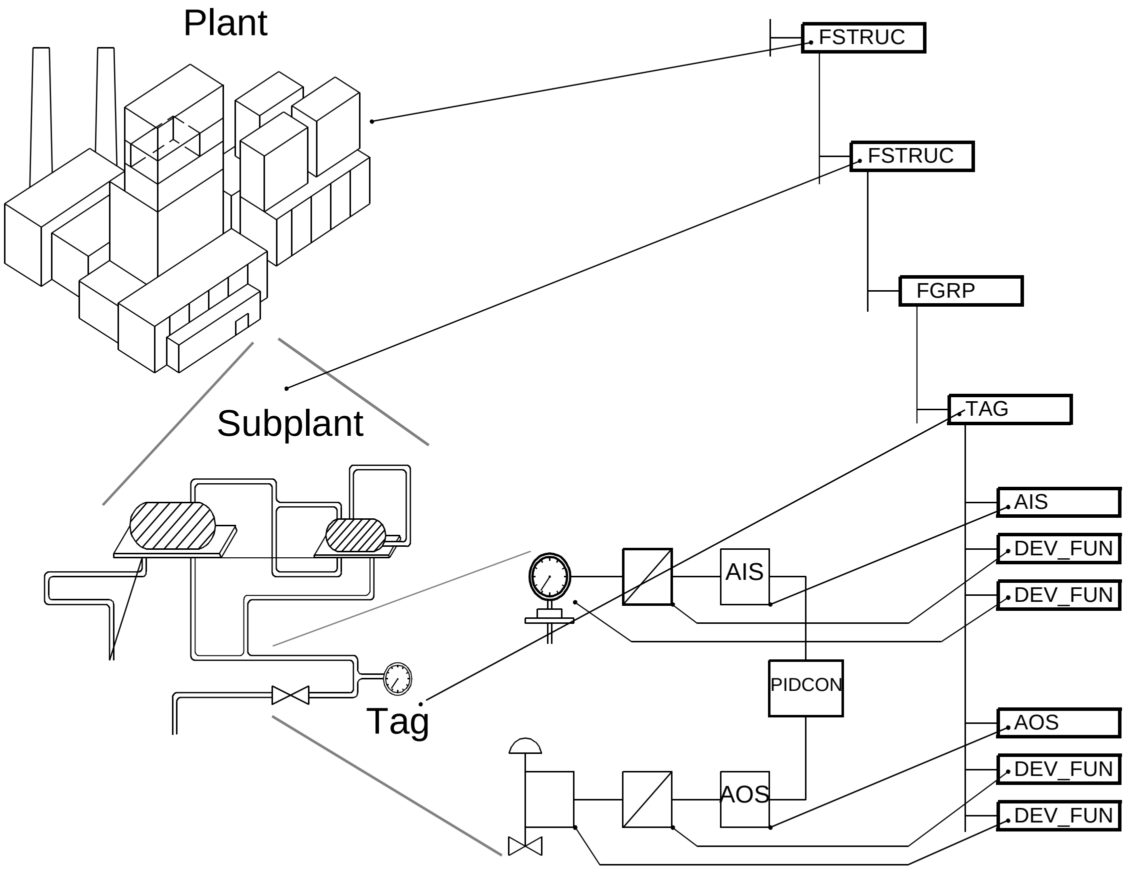


Figure 3-2. Plant/Function Oriented Structure

A separate substructure may be applied for each subject of delivery (for example, a substructure for the function oriented view for subject "instrumentation and control" and a separate substructure for the same main view for the subject "energy supply" within one project).

3.3.3 Location Oriented Structure

A location oriented structure is based on the topographical layout of the plant. A location oriented structure shows the subdivision of a location into constituent locations, without necessarily taking into account products and/or functions. The objects which represent locations are hierarchically positioned in this structure as shown in Figure 3-3.

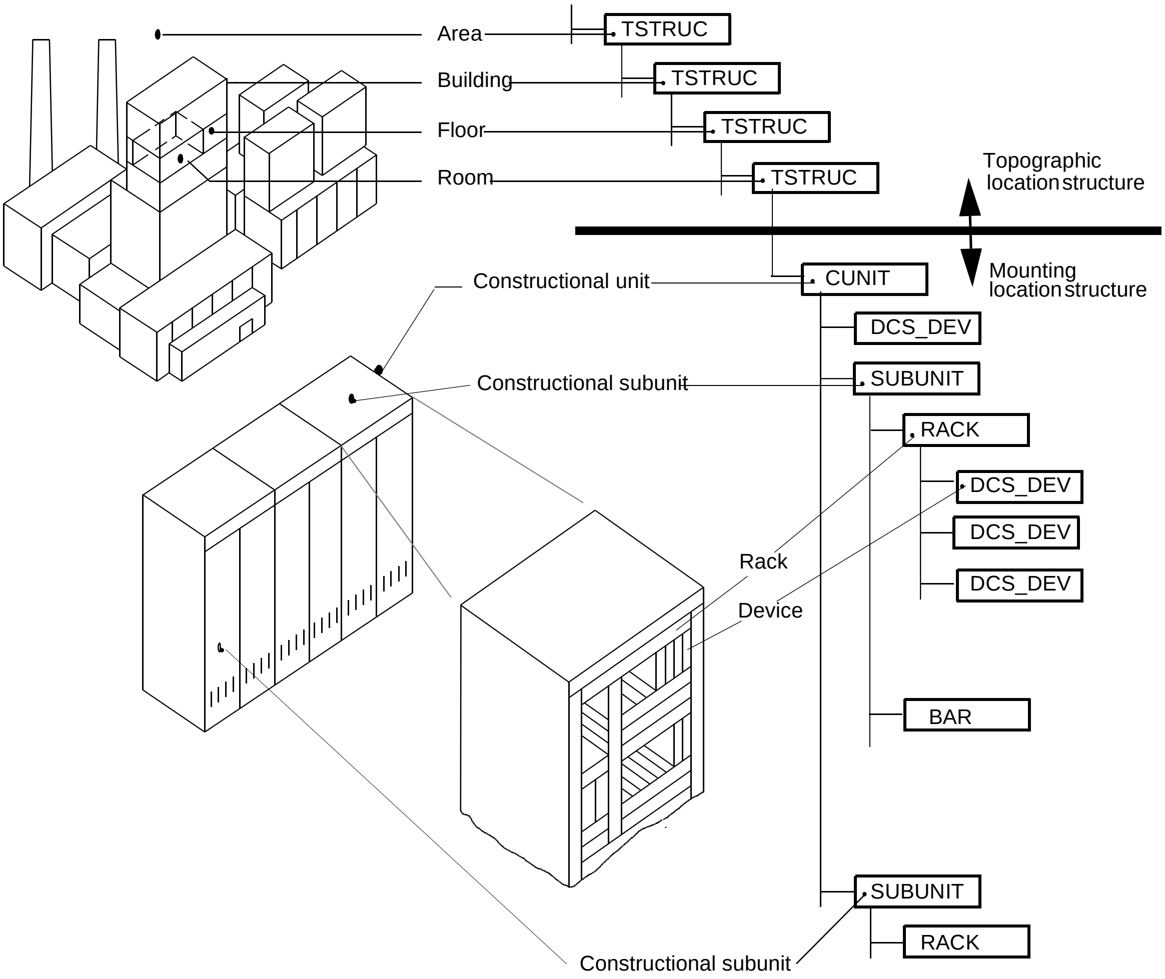


Figure 3-3. Location Oriented Structure

The illustration shows the border between the topographic location structure, in which the plant location objects are structured, and the mounting location structure, in which the constructional units like cabinets and desks are structured, further down to the mounting places for electrical and control equipment. Both types of location structure can also be applied independently of each other.

3.3.4 Control System Oriented Structure

The control system oriented structure shows the subdivision of a control system into constituent control system parts, without necessarily taking into account its functional, product, and locational environment within the plant where it is implemented. The objects which represent control system items are hierarchically positioned in this structure as shown in Figure 3-4.

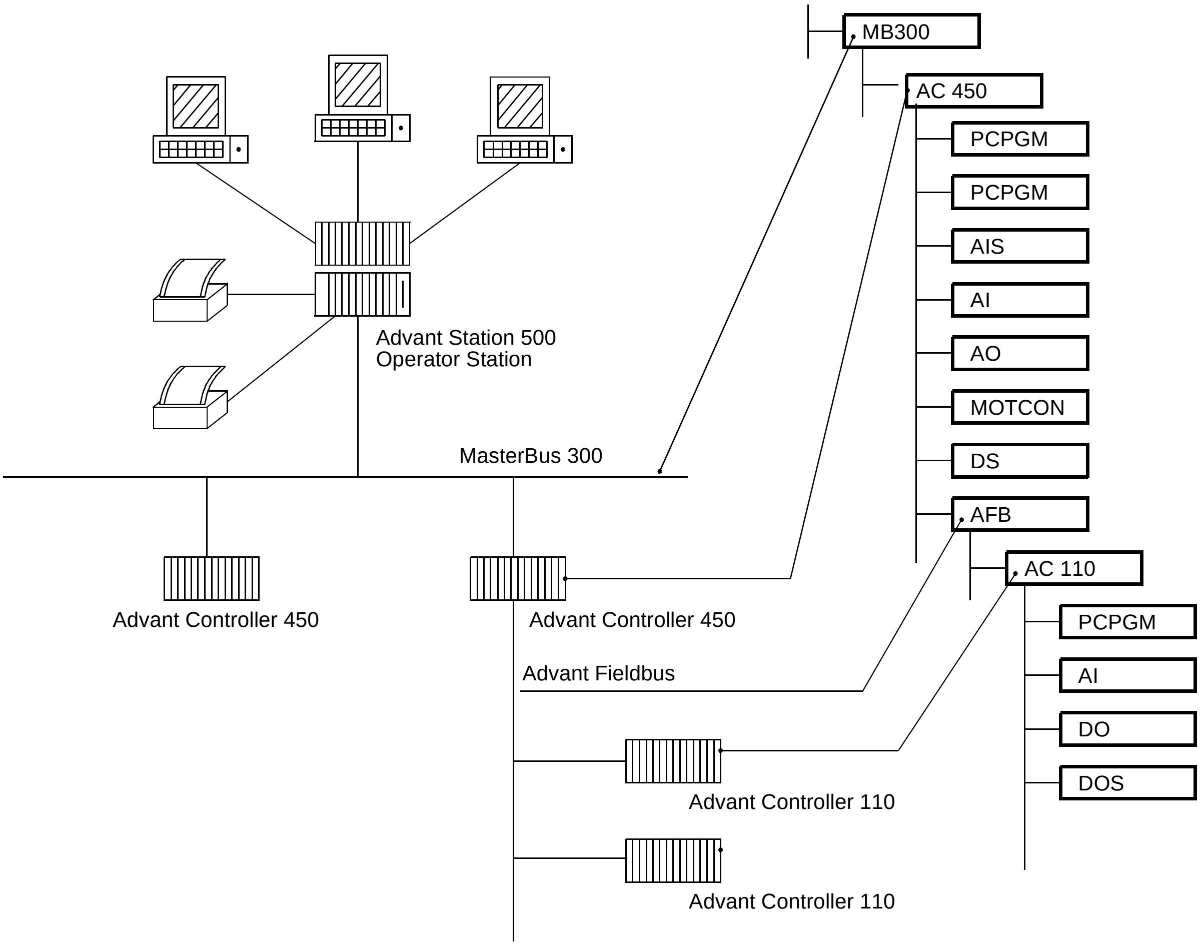


Figure 3-4. I Control System Oriented Structure

3.3.5 Plant Documentation Structure

The plant documentation structure specifies the subdivision of the plant documentation into documentation packages.

A documentation package is represented by an object which has a position in the documentation structure. The structure position determines the position of the documentation package in the overall plant documentation. The contents of a plant documentation package are determined by the relations of the documentation package to objects in the project structure.

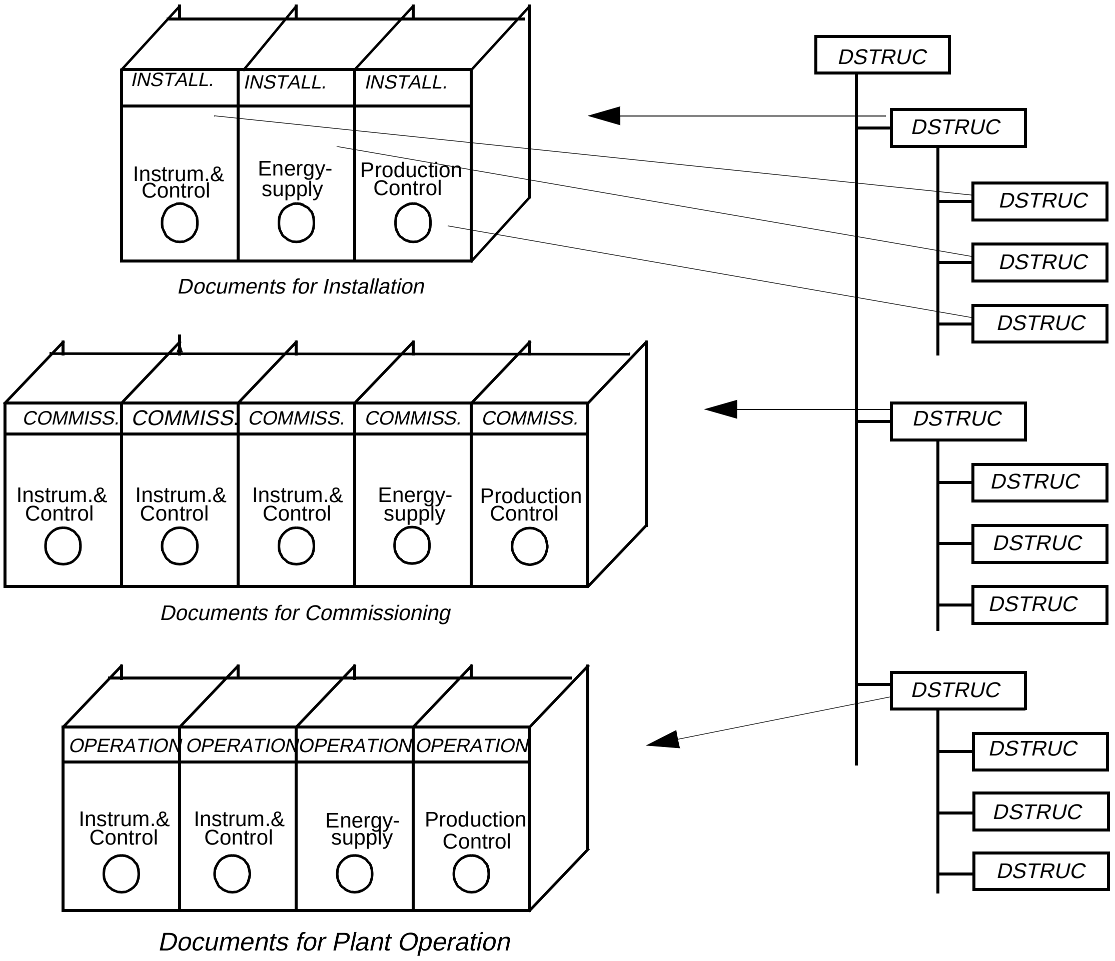


Figure 3-5. Plant Documentation Structure

The plant documentation oriented view is focused on the application of documentation (for installation, commissioning, or plant operation, for example) and on the subject of the delivery (documentation for the subject Instrumentation and Control, for example).

3.3.6 Relations between CAPE-objects

3.3.6.1 Links Between Objects

Objects within one structure type are related to each other by parent-child relationships, which are created automatically by building the structures. There are also relations between objects in different types of structures (for example, a DCS board element in the control system structure is implemented by a device-object which is located in an object of the location oriented structure), and relations other than parent-child relations between objects in the same structure. Such relations are established by creating links between the relevant objects. Within AdvaBuild there are several link-types defined and applicable. The link-type indicates a certain kind of relation between objects. Figure 3-6 provides an illustration of links between objects:

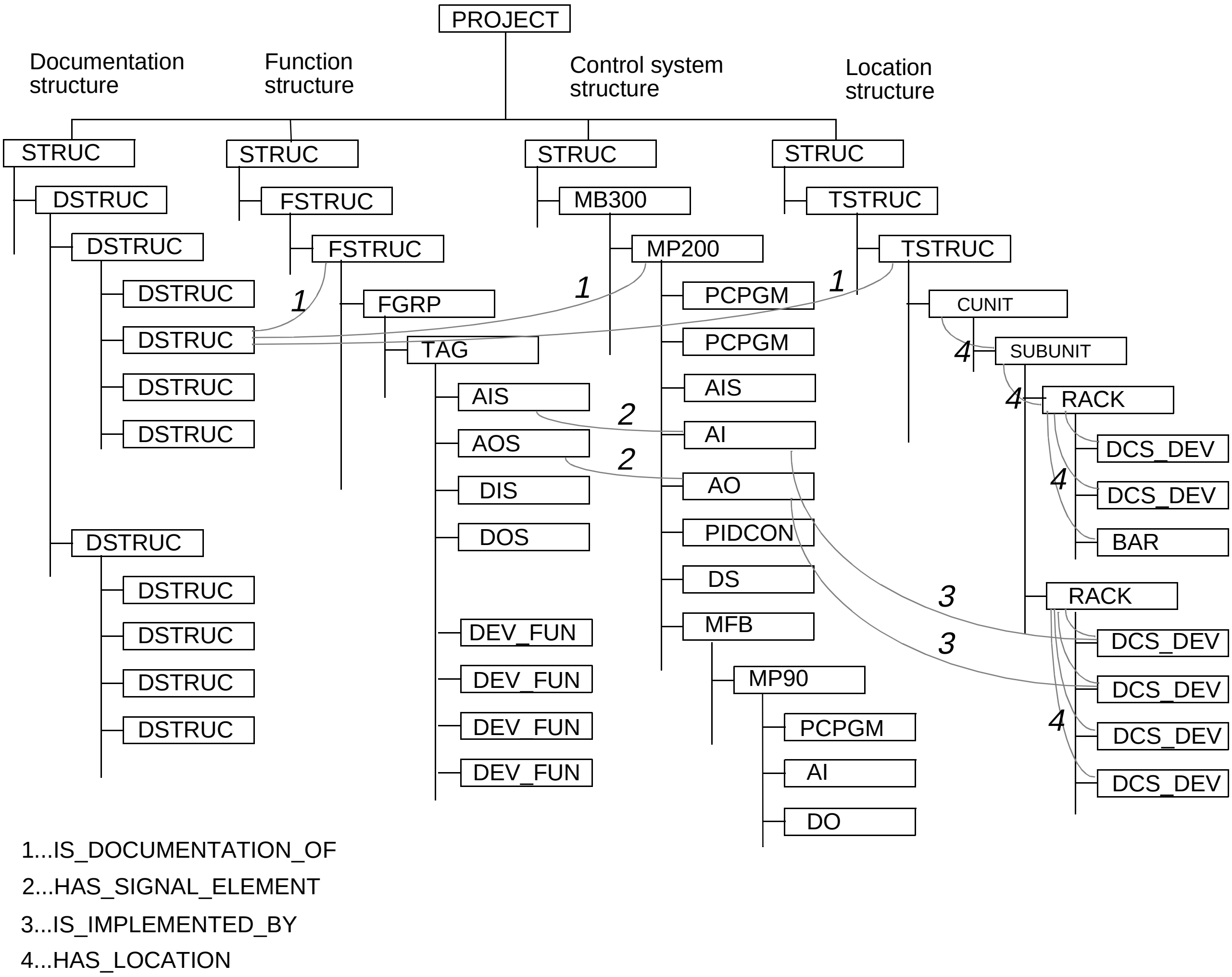


Figure 3-6. Links

3.3.6.2 Solutions (Engineering Units)

A solution is an object which represents a group of related objects that form an engineering unit. The objects which belong to a solution are linked to the solution object. The solution objects are put into a structure by its own called a solution pool. No hierarchies are foreseen in this structure in the present release. Objects which are linked to a solution may be positioned in any of the plant structure types. Figure 3-7 is an example of objects kept together by a solution object that specify and implement a tag.

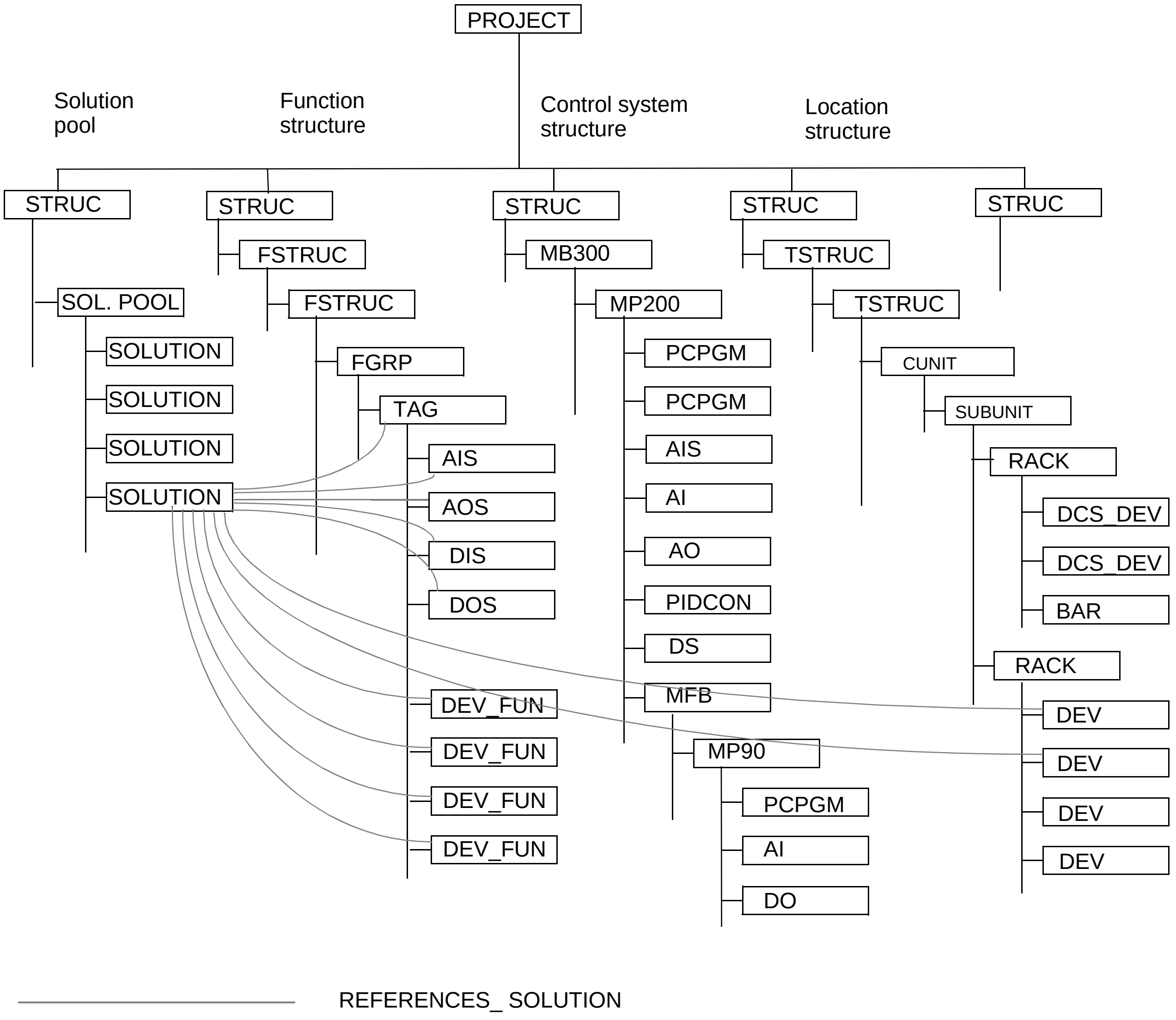


Figure 3-7. Objects of a Solution

Chapter 4 General Engineering Guidelines

4.1 Creation of the Plant Structures

4.1.1 Plant/Function Structure ("="-structure)

The plant/function structure (called function structure in AdvaBuild) is a reference structure that the process equipment, functions, and engineering are related to. The higher levels of this structure are based on the main equipment of a plant, whereas the lower levels are used to define the functions of sub-equipment (such as the instrumentation and control system).

This is the structure that function-objects are allocated to. It provides guidance for function oriented engineering activities.

This structure is built up as a tree structure in which the top object represents the functional view of the plant under consideration. Figure 4-1 shows an example of a function structure as presented by the Structure Builder:

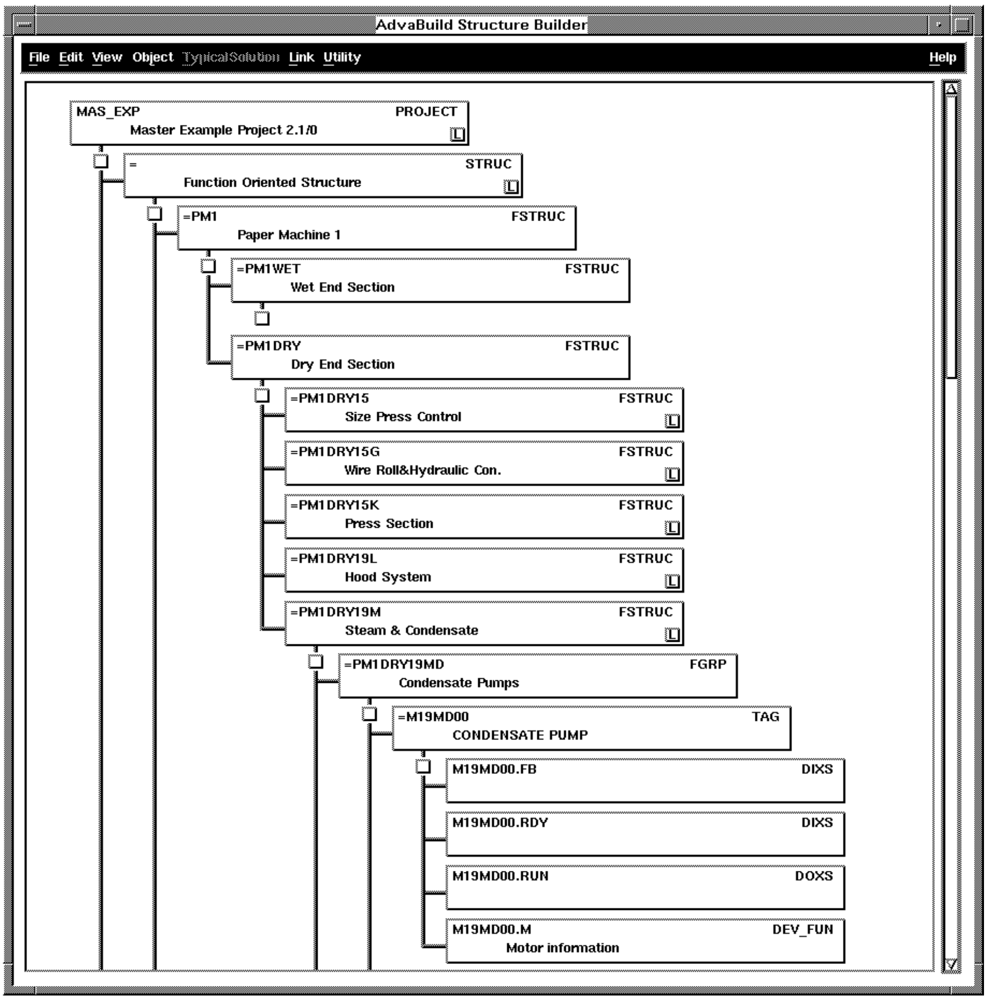


Figure 4-1. Function Structure

4.1.1.1 Object-types in the Function Structure

The function structure is built by AdvaBuild structure-objects. Table 4-1 and Table 4-2 show the object types which are defined for the function structure. These tables also show the associated plant objects that the object types represent and the maximum number of levels which can be structured by objects of the same object type.

Table 4-1 lists the object types which are used to build up the hierarchy of the function structure. These objects carry in their attributes little information concerning the plant/function objects which they represent. For more information, additional object types are assigned which represent information carrying document files. These object types representing document files are listed in Table 4-3.

Table 4-1. Structure Objects in the Function Structure

Object-type	Associated plant-objects	Parent-object	Max. number of levels
FSTRUC	Plant, sub-plant	STRUC, FSTRUC	4
FGRP	Functional group (control function)	FSTRUC	1
TAG	Measuring/actuating point	FSTRUC, FGRP	1

Table 4-2 lists the object types for the function structure, which carry in their attributes all the necessary information concerning the low-level plant/function objects which they represent:

Table 4-2. Signal Objects and Device Objects in the Function Structure

Object-type	Associated plant-objects	Parent-object	Max. number of levels
Object-types representing signal DB-elements (such as DIS, DOS, AIS, AOS)	Digital input signals, Digital output signals, Analog input signals, Analog output signals etc.	TAG	1
DEV_FUN	Device functions	TAG	1
DEV	Device	TAG	1

Table 4-3 lists the document types which describe objects in the function structure. These document types are represented by corresponding AdvaBuild object types, which can be assigned as children to the object-types in Table 4-1. They carry user defined information scopes concerning those objects to which they are assigned.

The information is presented as text document or as diagram. By assigning function and documentation-objects to structure-objects in the function structure, a parent-child link is automatically created between these objects:

Table 4-3. Documents in the Function Structure

Document-type	Object-type	Parent-object
Circuit diagram	CI_DIA	FSTRUC, FGRP, TAG
Free drawing	DRAWING	FSTRUC, FGRP, TAG
Assembly drawing	HOOKUP	FGRP, TAG
P&I diagram	PI_DIA	FSTRUC, FGRP
Free text	TEXT	FSTRUC, FGRP, TAG

Figure 4-2 shows examples of possible structure paths in the function structure:

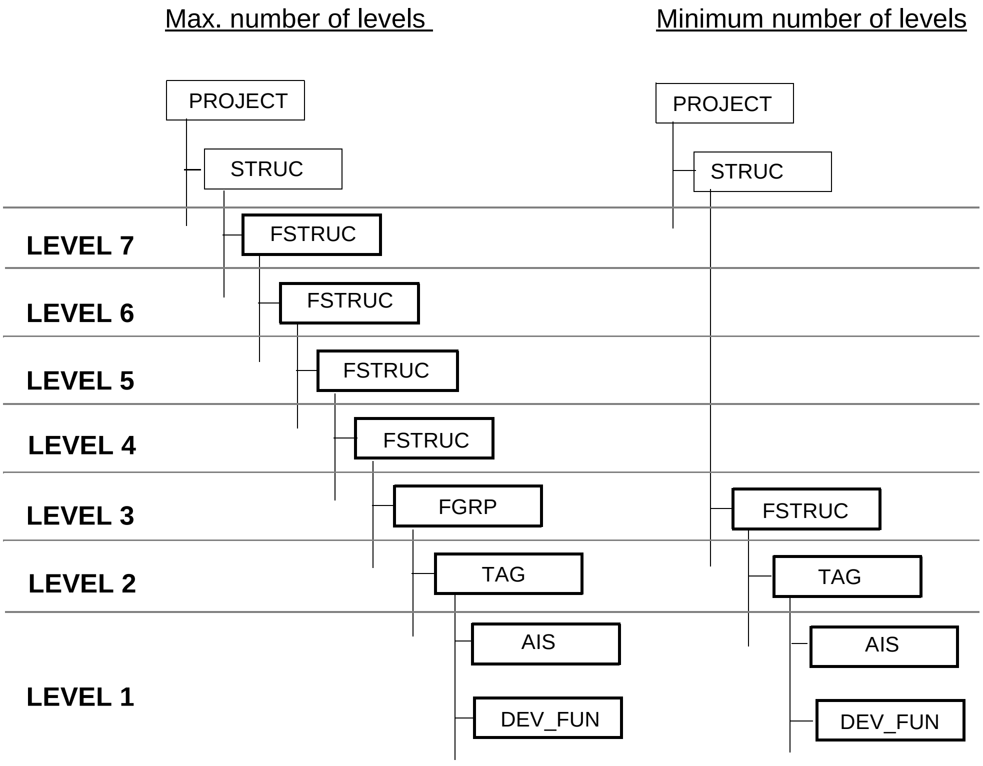


Figure 4-2. Structure Paths in the Function Structure

Figure 4-3 shows examples for assigning document-objects (object types representing document files) as children to function objects:

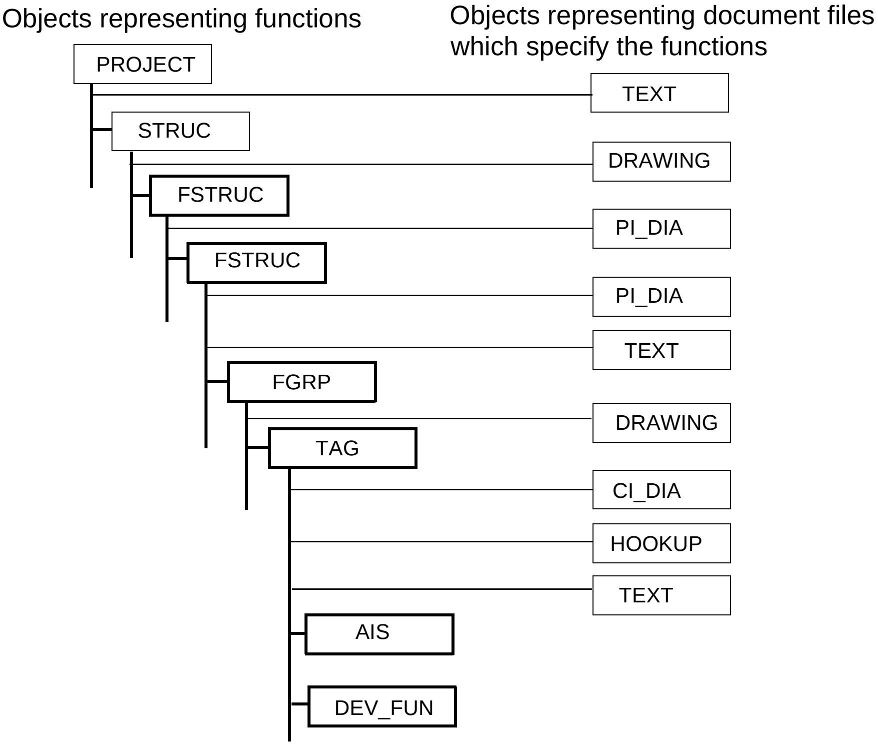

Objects representing functions
Objects representing document files which specify the functions
PROJECT
TEXT
STRUC
DRAWING
FSTRUC
PI_DIA
FSTRUC
PI_DIA
TEXT
FGRP
DRAWING
TAG
CI_DIA
HOOKUP
TEXT
AIS
DEV_FUN

Figure 4-3. Document-Objects assigned to Function-Objects

4.1.1.2 Link-types to Objects in the Function Structure

Relationships between objects which are not of the parent-child-type are defined by links. The kind of relationship is determined by the link-type. Objects outside the function structure can be linked to objects in the function structure by the following link-types:

Table 4-4. Links to Objects in the Function Structure

Link-type	Linked object-type	Example
IS_DOCUMENTATION_OF	DSTRUC to any object in the function structure	Links from plant documentation structure to objects in the function structure
HAS_SIGNAL_ELEMENT	Board element object to signal element object	Channel of analog input board to analog input signal
REFERENCES_SOLUTION	SOLUTION object in the solution pool to any object in the function structure	Links from a solution object to all objects that specify and implement a tag

4.1.1.3 Designation of Objects in the Function Structure

Objects in the function structure can be designated according to the rules which are stated in the standards IEC 750, DIN 40719 T2, and DIN 6779 (draft) for the designation block "higher level designation." The designation code can be used as object identity. The prefix "=" (equal) is used for function designations.

Figure 4-4 shows the object designation in a structure path with the maximum number of levels that are possible in AdvaBuild.

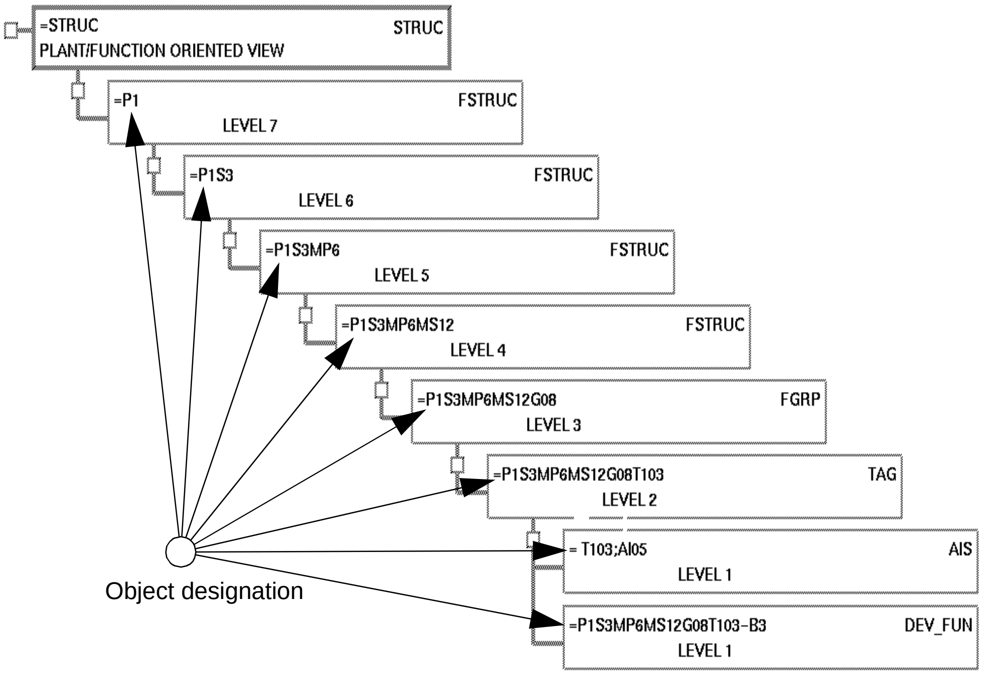


Figure 4-4. Designation of Objects in the Function Structure

Figure 4-5 shows the object designation in a structure path with the minimum number of levels used for a complete function structure:

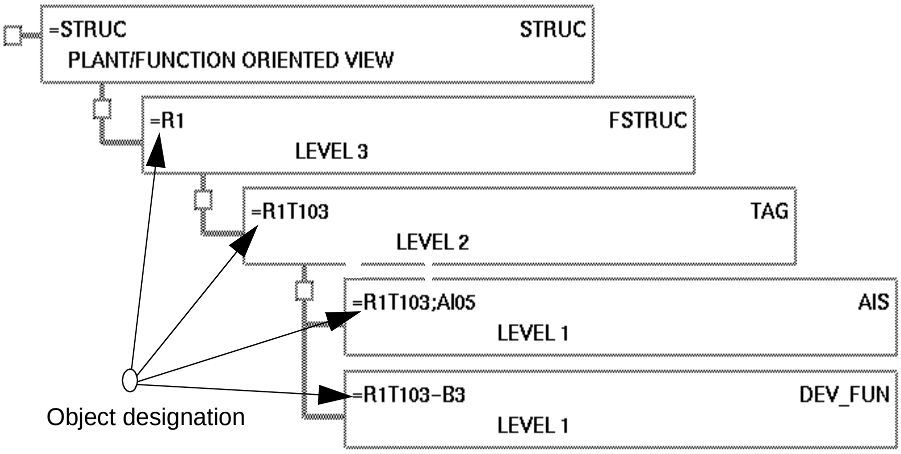


Figure 4-5. Designation of Objects in the Function Structure

Restrictions for object designation

The designations must be unique in the system and can have up to a maximum of 32 characters.

Signal designations used in the MASTER-system must not contain the "=" (equal) or ";" (semicolon) and must not exceed 20 characters.

The following rules apply according to the basic standards for designation:

- A transition between levels should be indicated in the designation by the transition from a numeric character to an alphabetic character.
- For the designation of tags, only the first letter code according to the ISO standard is used, followed by the tag number.
- The designation part for DCS-signal objects starts according to the standards with the prefix ";"(semicolon), but this is not allowed for use in the MASTER system.
- The designation part for devices and device-functions starts with the prefix "-" (minus).

4.1.1.4 Preconditions for the Design of the Function Structure

The function structure has to be designed in close cooperation with the customer. Normally the two highest levels are defined by the process engineers and represent the structuring into sub-plants.

The code which is used for designation must be agreed upon by the customer and other partners in the project. The level functional group/tag may be specified by or in cooperation with the partner who is responsible for process engineering. It is documented in P&I-diagrams and implemented by the AdvaBuild user according to that designation.

4.1.2 Topographic-location Structure ("++"-structure)

The topographic-location structure of a plant is a reference structure that the plant equipment is topographically related to. The structure is based on the topographic items of the plant such as areas, buildings, floors, rooms, and coordinates.

Within AdvaBuild this structure is used to allocate mounting locations (cabinets, desks, and so on) and components (stand alone equipment) to topographic locations (for example, assignment of a cabinet to a building, to the floor in the building, and to the coordinates within the floor).

The topographic-location structure is built up as a tree-like structure in which the top object represents the topographic view of the whole plant.

Figure 4-6 shows a topographic-location structure of a plant as presented by the Structure Builder:

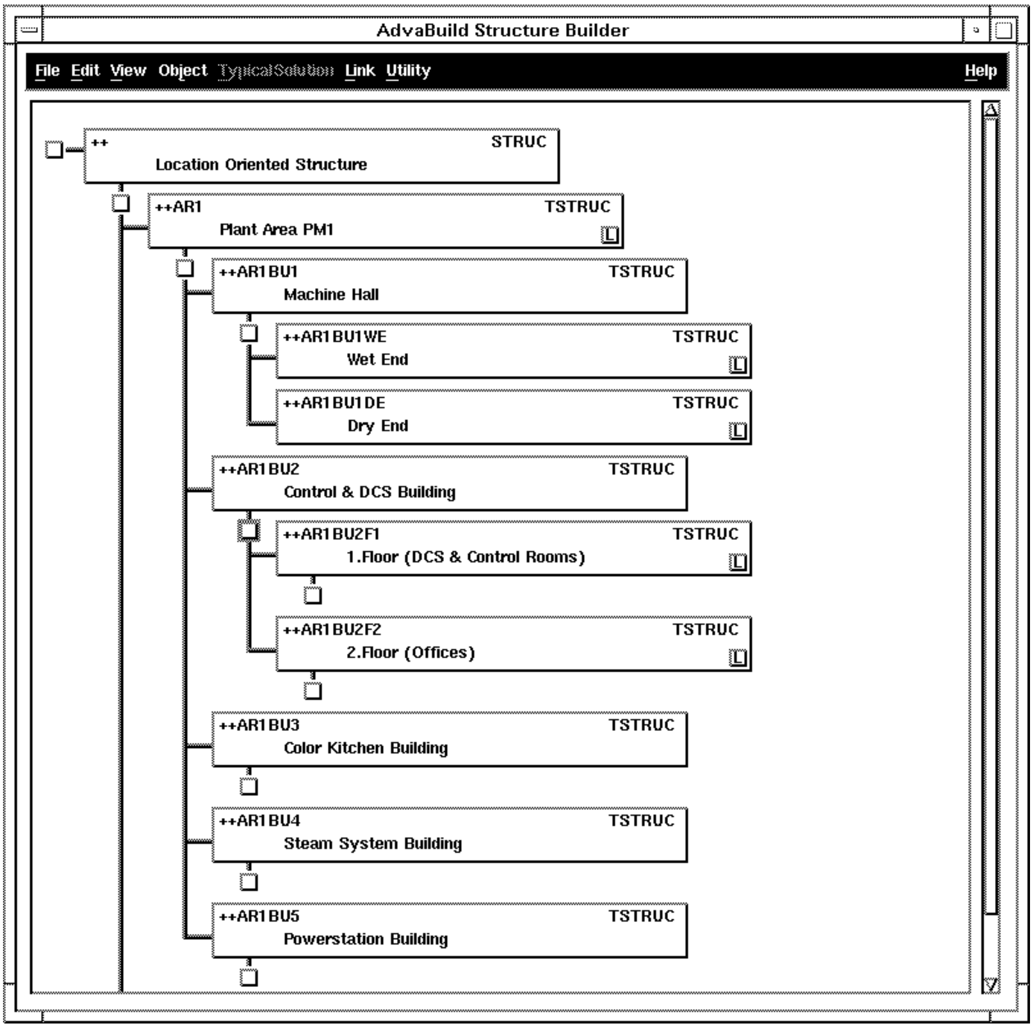


Figure 4-6. Topographic-Location Structure

4.1.2.1 Object-types in the Topographic-location Structure

The topographic-location structure is built by AdvaBuild structure-objects. For this structure the following object-type is defined:

Table 4-5. Structure Objects in the Topographic-location Structure

Object-type	Associated objects	Max. number of levels
TSTRUC	Topographic location within the plant	4

The topographic-location structure is continued into lower levels by the mounting-location structure. The following object-types represent mounting locations, and can be assigned as children to the structure-objects forming the topographic-location structure:

Table 4-6. Mounting-location Objects that can be assigned to the Topographic-location Structure

Object-type	Associated mounting-location objects
MSTRUC	Mounting location for electrical and control equipment
CUNIT	Cabinet or compound set of cabinets
SUBUNIT	Part of cabinet or part of compound cabinet set
RACK	Rack

Table 4-7 shows the document types which describe objects in the topographic location structure. These document types are represented by corresponding object types, which can be assigned as children to the object-type in Table 4-5. They carry user defined information scopes concerning those objects to which they are assigned.

Table 4-7. Documents in the Topographic-location Structure

Document-type	Object-type
Circuit diagram	CI_DIA
Free drawing	DRAWING
Arrangement drawing	LA_DIA
Free text	TEXT

4.1.2.2 Link-types to Objects in the Topographic-Location Structure

Relationships between objects which are not of the parent-child type are defined by links. The kind of relationship is determined by the link-type. Objects outside the topographic-location structure can be linked to objects in the topographic-location structure by the following link-types:

Table 4-8. Links to Objects in the Topographic-location Structure

Link-type	Linked object-type	Example
IS_DOCUMENTATION_OF	DSTRUC to TSTRUC	Links from plant documentation structure to TSTRUC-objects
REFERENCES_SOLUTION	SOLUTION object in the solution pool to any object in the topographic location structure	Link from a solution object to a room that contains electrical or control equipment

4.1.2.3 Designation of Objects in the Topographic-location Structure

Objects in the topographic-location structure can be designated according to the rules which are stated in the standards IEC 750, DIN 40719 T2, and DIN 6779 (draft) for the designation block "location designation" respective "topographic-location designation". This designation code is used as object identity.

Figure 4-7 shows the maximum number of levels that are possible in the topographic-location structure.

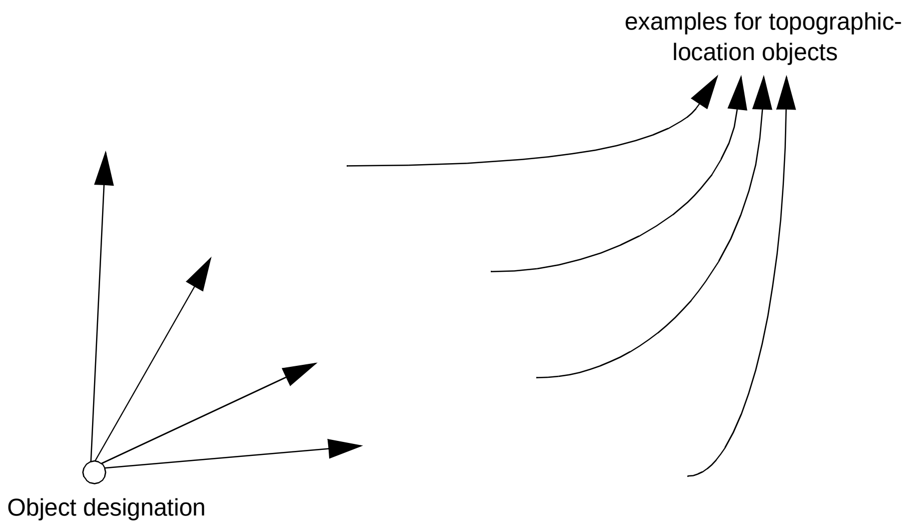


Figure 4-7. Designation of Objects in the Topographic-location Structure

In AdvaBuild the only restrictions for object designation are that the designation has to be unique in the system and the maximum number of characters must not exceed 32.

The following rule applies according to the basic standards for designation:

A transition between levels should be indicated in the designation by the transition from a numeric character to an alpha character.

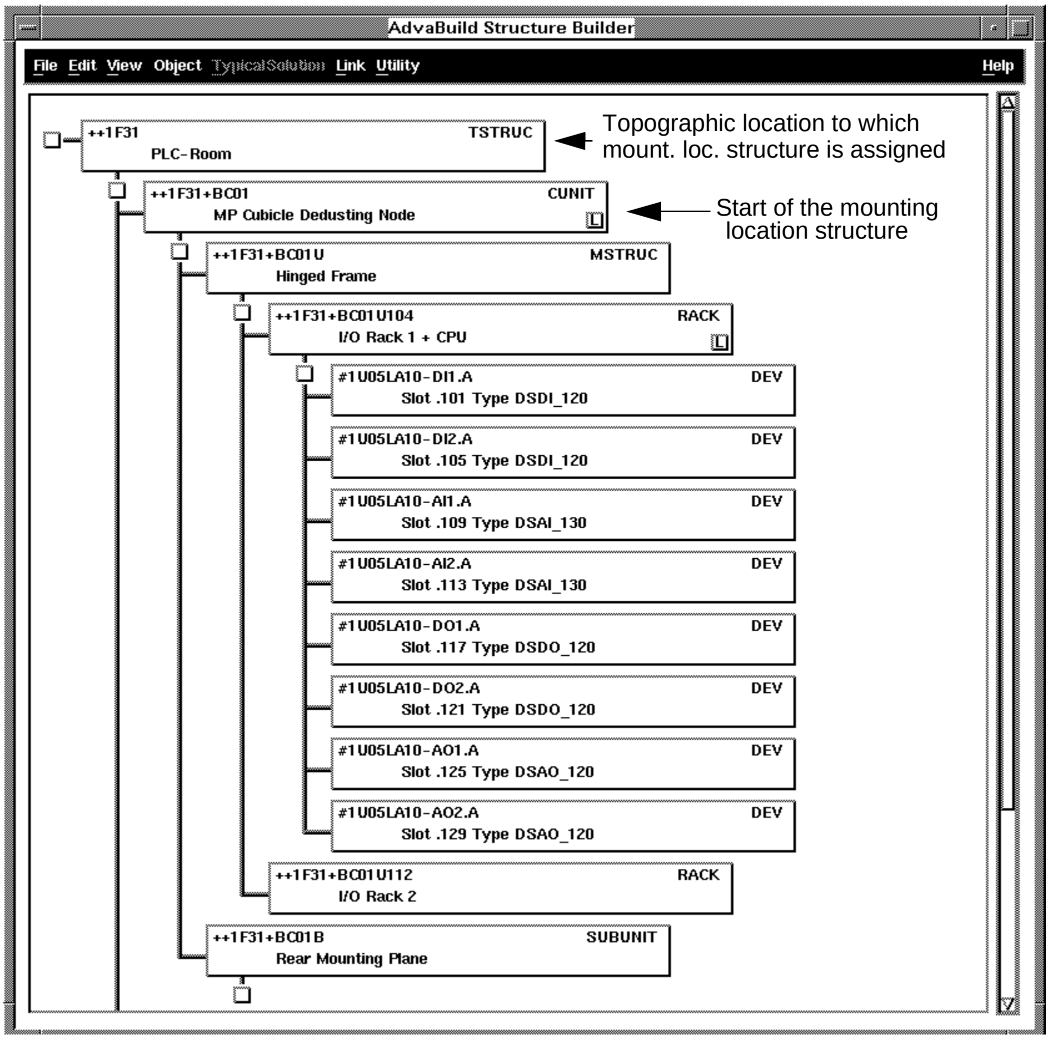
As already mentioned, AdvaBuild provides the continuation of the topographic-location structure into lower levels by objects of the mounting-location structure. Mounting-location objects are designated independently of the topographic-location structure. The designations of the topographic-location objects and of the mounting-location objects are concatenated when the structures are concatenated (see Section 4.1.3.3, Designation of Objects in the Mounting-location Structure). Both designations may be distinguished from each other by using the prefix "++" (double plus) for objects in the topographic-location structure and the prefix "+" (plus) for objects in the mounting-location structure.

4.1.2.4 Preconditions for the Design of the Topographic-location Structure

The topographic-location structure has to be defined in close cooperation with the partner who is responsible for civil engineering and plant layout. The relevant information which is documented in plant layouts and building drawings is evaluated by the AdvaBuild user and implemented as the topographic-location structure. The code used for the designation of the structure-objects has to be agreed upon by the customer and other partners in the project.

4.1.3 Mounting-location Structure ("+"-structure)

The mounting-location structure is the structure of locations and/or constructional units that are prepared to take up electrical and/or control equipment. Constructional units are cabinets, desks, racks, and so on, in which such equipment can be mounted. It is an independent structure which uniquely identifies the structure-objects within this structure. In AdvaBuild this structure is appended to the lower levels of the topographic-location structure. The engineering activities with objects of the mounting-location structure are not affected whether this structure is concatenated to the topographic-location structure or not. The top object of a mounting-location structure always represents a constructional unit or a mounting location. It may be created as an independent object and assigned to an object of the topographic-location structure in a later engineering activity, or it may be assigned to any object in the topographic-location structure and moved later to the final parent-object. Figure 4-8 shows a mounting-location structure as presented by the Structure Builder.



Topographic location to which mount. loc. structure is assigned

Start of the mounting location structure

Figure 4-8. Mounting-location Structure

4.1.3.1 Object-types in the Mounting-location Structure

The mounting-location structure is built by AdvaBuild structure-objects. For this structure the following object-types are defined in AdvaBuild:

Table 4-9. Objects in the Mounting-location Structure

Object-type	Associated objects	Parent object	Max. number of levels
MSTRUC	Mounting location	TSTRUC, CUNIT, SUBUNIT,	4
CUNIT	Cabinet or compound set of cabinets	TSTRUC, MSTRUC	1
SUBUNIT	Part of cabinet or part of compound cabinet set	MSTRUC, CUNIT	1
RACK	Mounting unit within cabinet	MSTRUC, CUNIT, SUBUNIT	1
BAR	Terminal bar	MSTRUC, CUNIT, SUBUNIT, RACK	1
DCS_DEV	Multi channel DCS-device	TSTRUC, MSTRUC, CUNIT,SUBUNIT, RACK	1
DEV	Device for general use	TSTRUC, MSTRUC, CUNIT,SUBUNIT, RACK	1
CABLE	Cable	TSTRUC, MSTRUC, CUNIT, SUBUNIT, RACK,	1

Table 4-10 shows the document types which describe objects in the mounting-location structure. These document types are represented by corresponding AdvaBuild object types, which can be assigned as children to the relevant object-types in Table 4-9. They carry user defined information scopes concerning those objects to which they are assigned.

Table 4-10. Documents in the Mounting-location Structure

Document-type	Object-type	Parent-object
Circuit diagram	CI_DIA	MSTRUC, CUNIT, SUBUNIT, RACK
Free drawing	DRAWING	MSTRUC, CUNIT, SUBUNIT, RACK, BAR
Arrangement drawing	LA_DIA	MSTRUC, CUNIT, SUBUNIT, RACK
Free text	TEXT	CUNIT, SUBUNIT, RACK, BAR

4.1.3.2 Link-types to Objects in the Mounting-location Structure

Relationships between objects which are not of the parent-child type are defined by links. The kind of relationship is determined by the link-type. Objects outside the mounting-location structure can be linked to objects in the mounting-location structure by the following link-types:

Table 4-11. Links to Objects in the Mounting-location Structure

Link-type	**Linked object-type**	**Example**
IS_DOCUMENTATION_OF	DSTRUC to any object in mounting-location structure	Link from a defined documentation package in the plant documentation structure to an object in the mounting-location structure
IS_IMPLEMENTED_BY	DCS-board objects to DEV_FUN objects	Link from a digital input board in the control system structure to its implementation in the mounting-location structure
REFERENCES_SOLUTION	SOLUTION object in the solution pool to any object in the mounting location structure	Links from a solution object to all objects that implement a tag

Objects of the mounting-location structure can also be linked with other objects in the same structure by the link-type HAS_LOCATION. These links are assigned as open links to objects that have mounting locations defined. They are closed by assigning equipment to these mounting locations. An open HAS_LOCATION link defines an unoccupied mounting location.

Table 4-12. Links to Objects in the Mounting-location Structure with other objects

Link-type	**Linked object-type**
HAS_LOCATION	TSTRUC to DCS_DEV, DEV, BAR MSTRUC to DCS_DEV, DEV, BAR CUNIT to DCS_DEV, DEV, BAR SUBUNIT to DCS_DEV, DEV, BAR RACK to DCS_DEV, DEV, BAR

A closed HAS_LOCATION link defines an occupied mounting location.

4.1.3.3 Designation of Objects in the Mounting-location Structure

Objects in the mounting-location structure can be designated according to the rules which are stated in the standards IEC 750, DIN 40719 T2, and DIN 6779 (draft) for the designation block "location designation" respective "mounting-location designation". The designation code is used as object identity.

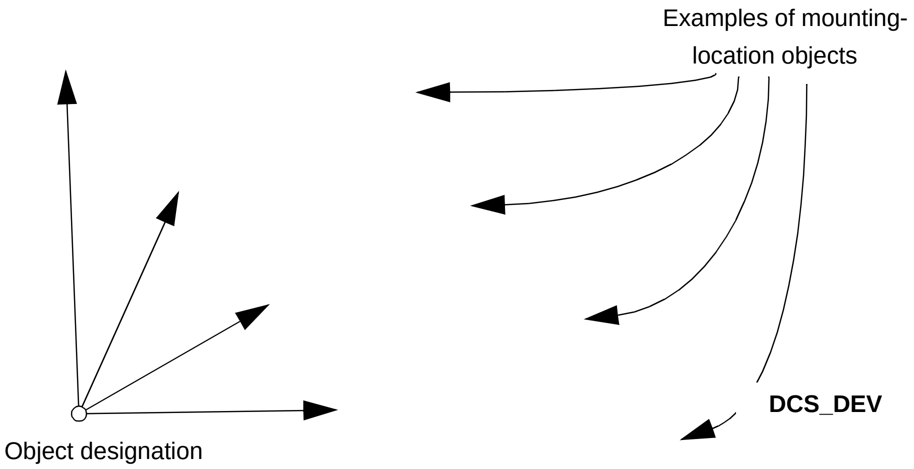


Figure 4-9. Designation of Objects in the Mounting-Location Structure

The designation levels in the mounting-location structure correspond to the structure levels. The only restrictions for object designation are that the designation has to be unique in the project and the maximum number of characters may not exceed 32.

However, the following rules must be applied according to the basic standards for designation:

- A transition between levels should be indicated in the designation by the transition from a numeric character to an alpha character.
- The designation part for DCS-devices and device-functions starts with the prefix "-" (minus).

The final object designation of a mounting-location object is a combination of the designation of the topographic-location object to which the mounting-location object is assigned as child and the designation (object id) of the mounting-location object.

4.1.3.4 Preconditions for the Design of the Mounting-location Structure

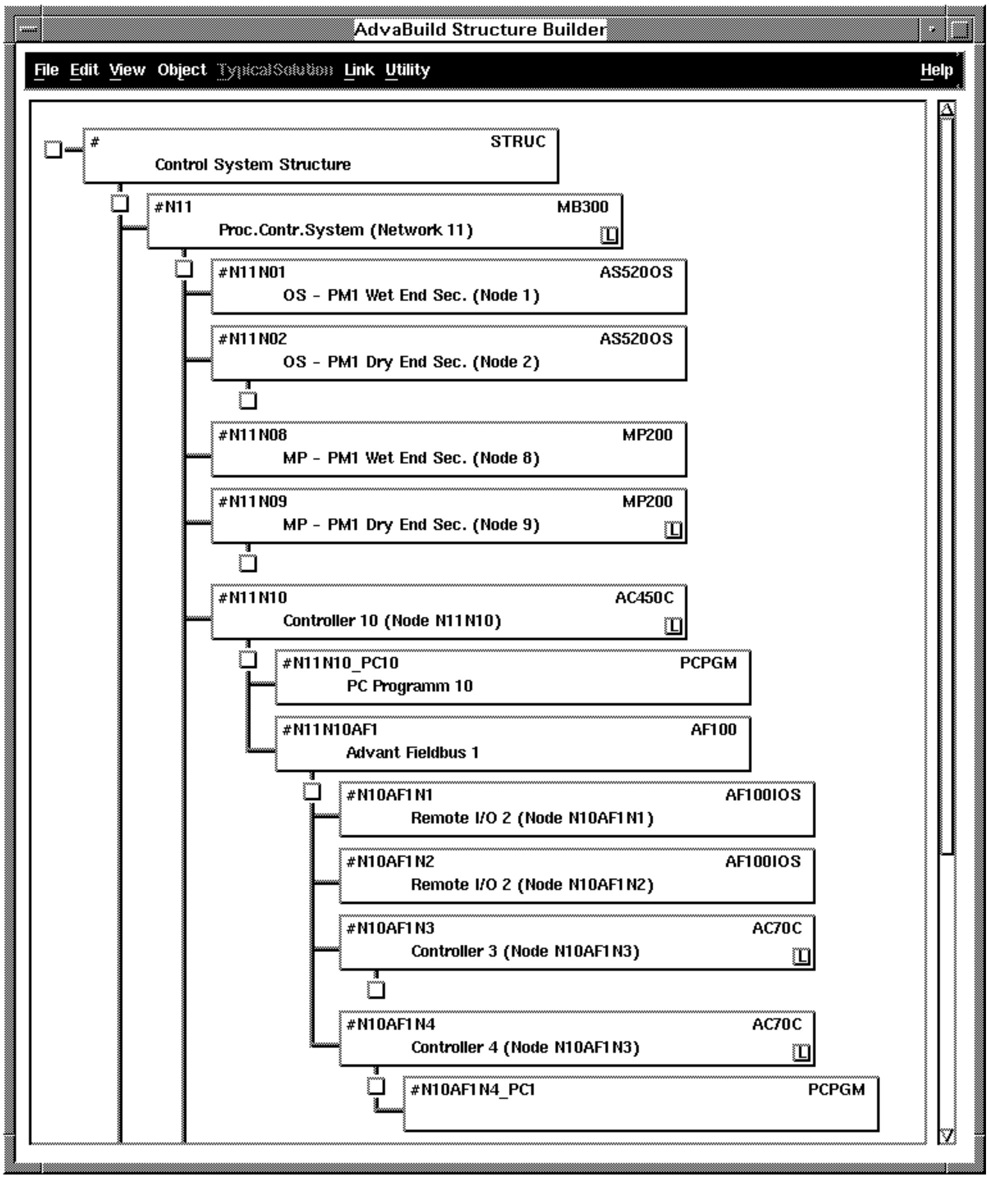
The user is free to create this structure according to the needs for locating and interconnecting the electrical and control equipment.

4.1.4 Control System Structure ("#"-structure)

The control system structure is a reference structure to which the system functions of the control system and the application functions are related.

This structure can be built up as a tree structure in which the top object represents the view of the control system.

Figure 4-10 shows an example of a control system structure as presented by the Structure Builder:



STRUC Control System Structure
#N11 MB300 Proc.Contr.System (Network 11)
#N11N01 AS520OS OS - PM1 Wet End Sec. (Node 1)
#N11N02 AS520OS OS - PM1 Dry End Sec. (Node 2)
#N11N08 MP200 MP - PM1 Wet End Sec. (Node 8)
#N11N09 MP200 MP - PM1 Dry End Sec. (Node 9)
#N11N10 AC450C Controller 10 (Node N11N10)
#N11N10_PC10 PCPGM PC Programm 10
#N11N10AF1 AF100 Advant Fieldbus 1
#N10AF1N1 AF100IOS Remote I/O 2 (Node N10AF1N1)
#N10AF1N2 AF100IOS Remote I/O 2 (Node N10AF1N2)
#N10AF1N3 AC70C Controller 3 (Node N10AF1N3)
#N10AF1N4 AC70C Controller 4 (Node N10AF1N3)
#N10AF1N4_PC1 PCPGM

Figure 4-10. Control System Structure

4.1.4.1 Object-types in the Control System Structure

The control system structure is built up by AdvaBuild defined structure-objects. The object-types in this structure are dependent on the target-system. Examples for object-types in the control function structure for a MASTER DCS-system are shown in the following table:

Table 4-13. Examples of Objects in the Control System Structure

Object-type	Associated DCS objects
MB300	Master Bus 300
MP200	Master Piece 200/1
AC110C	Advant Controller 110
AC410C	Advant Controller 410
AC450C	Advant Controller 450
MV850	Master View 850
DISPGRP	Group of graphical displays.
DISP	Graphical display for process supervision and control via AdvaCommand User Interface
PCPGM	Process control program.
PIDCON	PID-controller functional unit.
AI	Analog input board element.
DI	Digital input board element.

Note that the above listed objects are only examples and not a complete list of object-types in the control system structure. There are far more object-types defined for the MASTER control systems (see the *AdvaBuild Object Types Reference Manual*).

Table 4-14 shows the document types which describe objects in the control system structure. These document types are represented by corresponding AdvaBuild object types, which can be assigned as children to the relevant object-types in Table 4-13. They carry user defined information scopes concerning those objects to which they are assigned.

Table 4-14. Documents in the Control System Structure

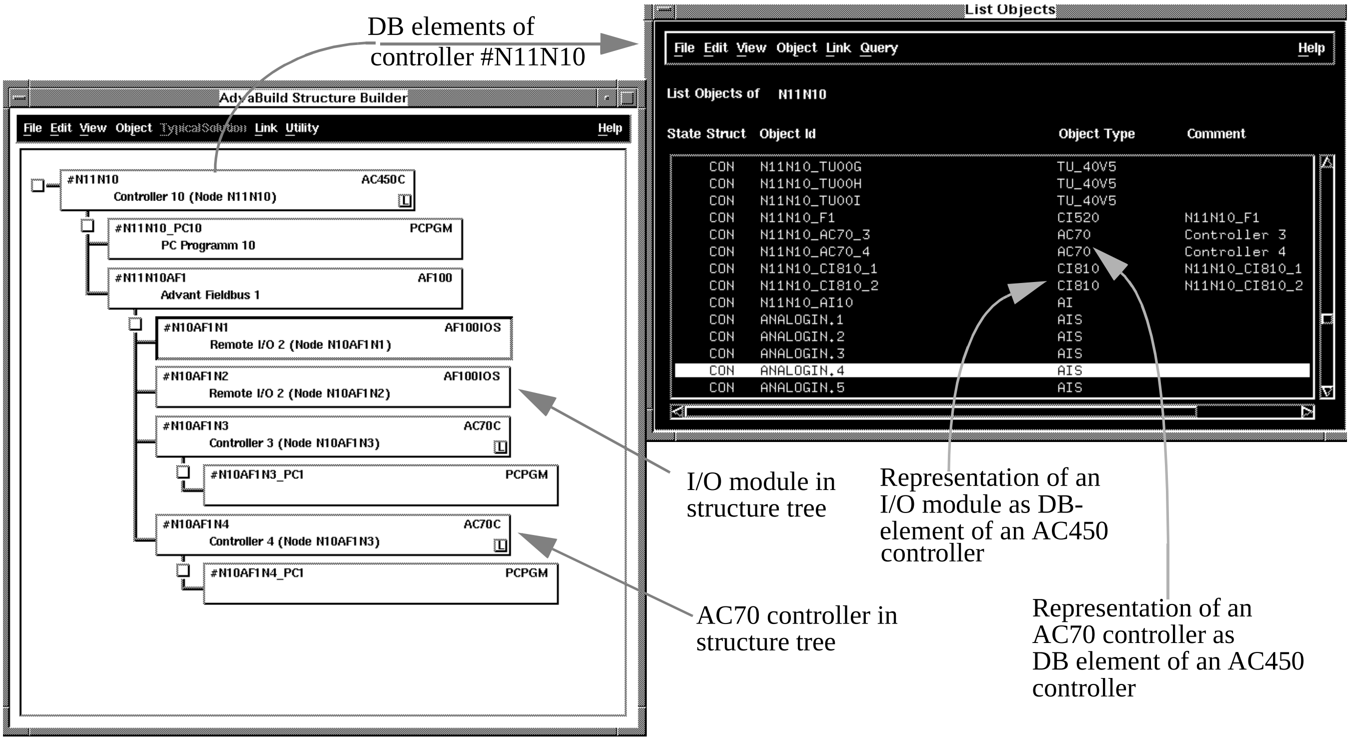
Document-type	object-type	parent-object
Free Drawing	DRAWING	MB300, MP200, AC410C/450C, PCPGM,...
Free text	TEXT	MB300, MP200, AC410C/450C, PCPGM,...

4.1.4.2 Special Hints for the Creation of the Control System Structure

Some objects have two representations in the control system structure. Such objects are for example:

- Advant Controller AC70
  - Object type representing the DB element:AC70
  - Object type for presentation in tree structure: AC70C
- Advant Controller AC110
  - Object type representing the DB element:AC110
  - Object type for presentation in tree structure: AC110C
- Advant Fieldbus 100
  - Object type representing the DB element: CI520
  - Object type for presentation in tree structure: AF100
- S800 I/O Module Elements
  - Object type representing the DB element: CI810
  - Object type for presentation in tree structure: AF100IOS.

The DB element representation must be assigned as child to a controller object in the form of a list object thus representing the assignment to a specific controller. The other representation is in the form of a tree object in the control system structure. Children of such objects are assigned to its representation in the tree. Figure 4-11 shows examples of I/O module elements and of AC70 controllers in the control system structure.


DB elements of controller #N11N10
AdvaBuild Structure Builder
File Edit View Object TypicalSolution Link Utility Help
#N11N10 AC450C
Controller 10 (Node N11N10)
#N11N10_PC10 PCPGM
PC Programm 10
#N11N10AF1 AF100
Advant Fieldbus 1
#N10AF1N1 AF100IOS
Remote I/O 2 (Node N10AF1N1)
#N10AF1N2 AF100IOS
Remote I/O 2 (Node N10AF1N2)
#N10AF1N3 AC70C
Controller 3 (Node N10AF1N3)
#N10AF1N3_PC1 PCPGM
#N10AF1N4 AC70C
Controller 4 (Node N10AF1N3)
#N10AF1N4_PC1 PCPGM
List Objects
File Edit View Object Link Query Help
I/O module in structure tree
AC70 controller in structure tree
Representation of an I/O module as DB-element of an AC450 controller
Representation of an AC70 controller as DB element of an AC450 controller

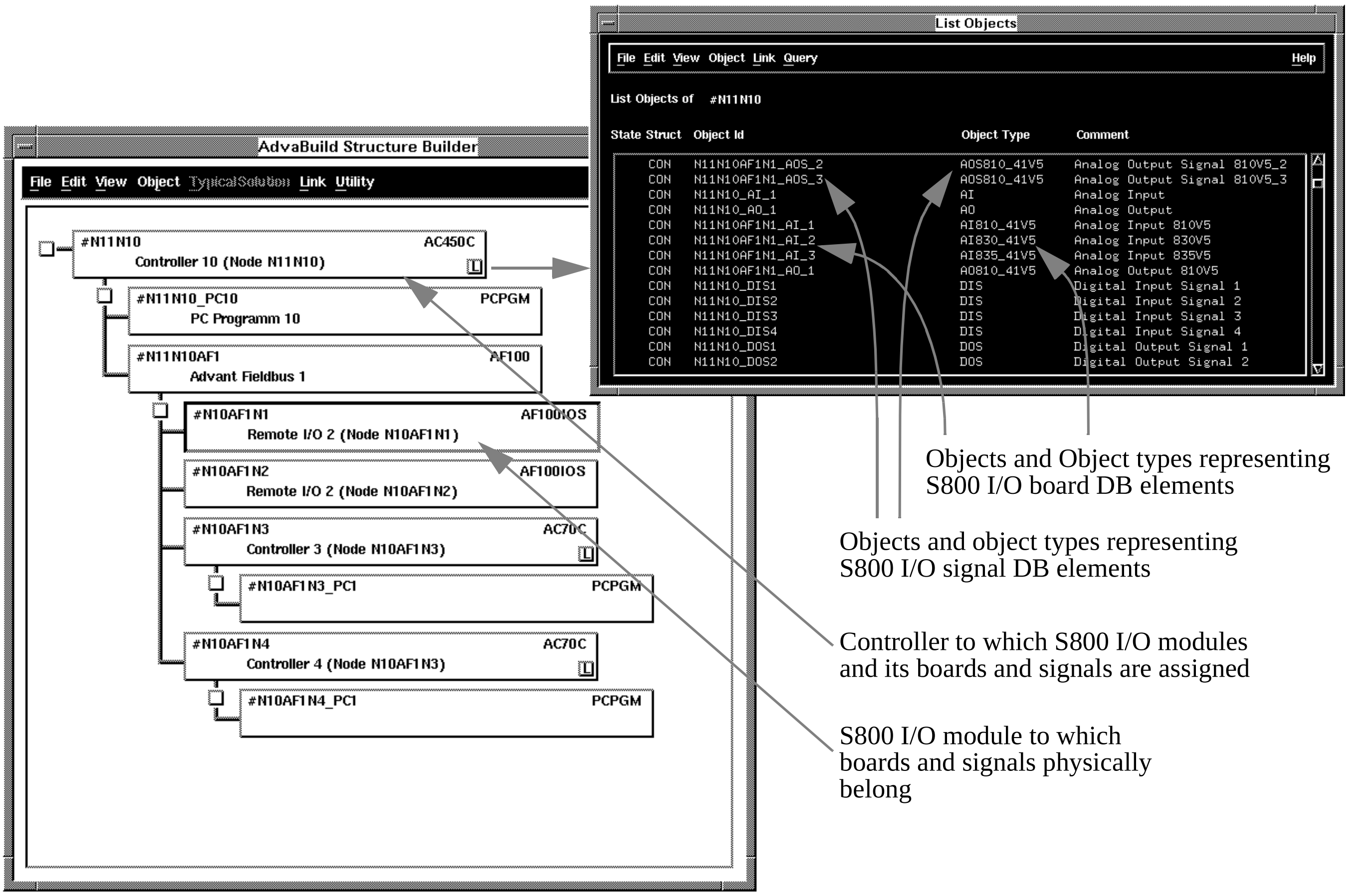
```
List Objects of   N11N10

State Struct  Object Id               Object Type   Comment

     CON   N11N10_TU00G              TU_40V5
     CON   N11N10_TU00H              TU_40V5
     CON   N11N10_TU00I              TU_40V5
     CON   N11N10_F1                 CI520         N11N10_F1
     CON   N11N10_AC70_3             AC70          Controller 3
     CON   N11N10_AC70_4             AC70          Controller 4
     CON   N11N10_CI810_1            CI810         N11N10_CI810_1
     CON   N11N10_CI810_2            CI810         N11N10_CI810_2
     CON   N11N10_AI10               AI
     CON   ANALOGIN.1                AIS
     CON   ANALOGIN.2                AIS
     CON   ANALOGIN.3                AIS
     CON   ANALOGIN.4                AIS
     CON   ANALOGIN.5                AIS
```

Figure 4-11. Placing of Objects in the Control System Structure

The object types AF100IOS representing S800 I/O Module Elements are only used for presentation purpose in the tree structure. The I/O board objects and signal objects belonging to such I/O stations must be assigned directly to that controller object to which the I/O station is assigned. Figure 4-12 shows examples of S800 I/O boards and signals assigned to a controller


AdvaBuild Structure Builder
File Edit View Object TypicalSolution Link Utility
#N11N10 AC450C
Controller 10 (Node N11N10)
#N11N10_PC10 PCPGM
PC Programm 10
#N11N10AF1 AF100
Advant Fieldbus 1
#N10AF1N1 AF100IOS
Remote I/O 2 (Node N10AF1N1)
#N10AF1N2 AF100IOS
Remote I/O 2 (Node N10AF1N2)
#N10AF1N3 AC70C
Controller 3 (Node N10AF1N3)
#N10AF1N3_PC1 PCPGM
#N10AF1N4 AC70C
Controller 4 (Node N10AF1N3)
#N10AF1N4_PC1 PCPGM
List Objects
File Edit View Object Link Query Help

List Objects of #N11N10

State Struct	Object Id	Object Type	Comment
CON	N11N10AF1N1_AOS_2	AOS810_41V5	Analog Output Signal 810V5_2
CON	N11N10AF1N1_AOS_3	AOS810_41V5	Analog Output Signal 810V5_3
CON	N11N10_AI_1	AI	Analog Input
CON	N11N10_AO_1	AO	Analog Output
CON	N11N10AF1N1_AI_1	AI810_41V5	Analog Input 810V5
CON	N11N10AF1N1_AI_2	AI830_41V5	Analog Input 830V5
CON	N11N10AF1N1_AI_3	AI835_41V5	Analog Input 835V5
CON	N11N10AF1N1_AO_1	AO810_41V5	Analog Output 810V5
CON	N11N10_DIS1	DIS	Digital Input Signal 1
CON	N11N10_DIS2	DIS	Digital Input Signal 2
CON	N11N10_DIS3	DIS	Digital Input Signal 3
CON	N11N10_DIS4	DIS	Digital Input Signal 4
CON	N11N10_DOS1	DOS	Digital Output Signal 1
CON	N11N10_DOS2	DOS	Digital Output Signal 2

Objects and Object types representing S800 I/O board DB elements

Objects and object types representing S800 I/O signal DB elements

Controller to which S800 I/O modules and its boards and signals are assigned

S800 I/O module to which boards and signals physically belong

Figure 4-12. S800 I/O Board and Signal Objects in the Control System Structure

4.1.4.3 Link-types to Objects in the Control System Structure

Relations between objects in the control system structure which are not of the parent-child type to objects in other AdvaBuild structures or to other objects in the control system structure are defined by the following link-types:

Table 4-15. Links to Objects in the Control System Structure

Link-type	Linked object-type	Example
IS_DOCUMENTATION_OF	DSTRUC to any object in the control system structure.	Links from plant documentation structure to objects in the control system structure.
HAS_SIGNAL_ELEMENT	DCS-board element objects to DCS signal element objects.	Links from channel of analog input board to analog input signal.
IS_IMPLEMENTED_BY	DCS-board element objects to DCS_DEV.	Links from analog input board to implemented analog input device in the mounting-location structure
REFERENCES_SOLUTION	SOLUTION object in the solution pool to any object in the control system structure.	Links from a solution object to objects that belong to an engineering unit.

4.1.4.4 Designation of Objects in the Control System Structure

The objects in the control system structure are designated with a code, which is presented in a designation block with the prefix "#". This code represents the concatenating of the object codes of the objects within a structure path from the top object down to the object of interest. There are no standards in use for the designation of a control system structure.

Figure 4-13 shows an example of the designation of control system objects:

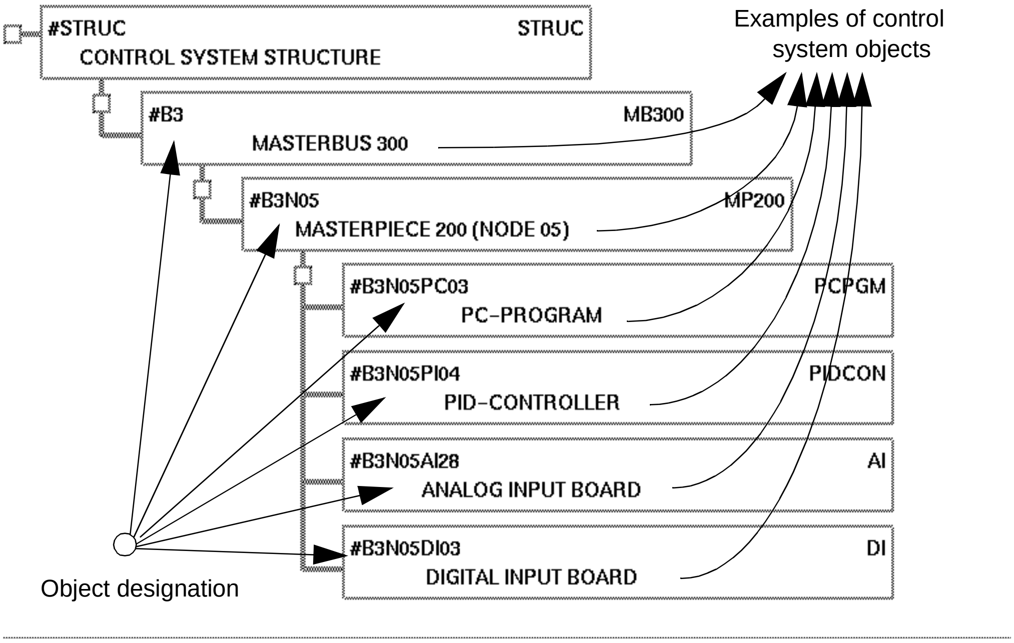


Figure 4-13. Designation of Objects in the Control System Structure

The designation has to be unique in the project and the maximum number of characters may not exceed 32.

Signal designations used in the MASTER-system must not contain the "=" (equal) and ";" (semicolon) signs and must not exceed 20 characters.

A transition between levels should be indicated in the designation by the transition from a numeric character to an alpha character.

4.1.4.5 Preconditions for the Design of the Control System Structure

A precondition for the design of the control system structure is an estimation of the scope of control functions, their distribution within the control system, and their runtime requirements. The peripheral equipment and the signals which are to be connected to the control system must be estimated. Also, the requirements with regard to the location of control system elements must be specified.

These estimations are done by the user and should be documented in text- or diagram-objects.

4.1.5 Plant Documentation Structure ("&"-structure)

The plant documentation has to be structured according to the plant structures ("="-, "+"-, "#"-structures) and according to:

- application of documentation
- subjects of delivery.

The structure of documentation output can be preconfigured in AdvaBuild by creating a documentation structure, in which the DSTRUC-objects represent documentation packages structured, for example, according to subjects, applications, and plant sections. It is predefined which document-types belong to distinct subjects and applications. The scope of the acquired documents within a documentation package is defined by links to the structures in which the documents are created.

The plant documentation structure is built up as a tree-like structure. The top structure-object represents the complete scope of plant documentation according to the documentation planning. The objects in the following levels represent documentation packages for distinct applications and subjects. There should only be three levels created. The content of each package depends on its linking to objects in the function-, location-, and control system structures.

Figure 4-14 shows an example of a plant documentation structure as presented by the Structure Builder:

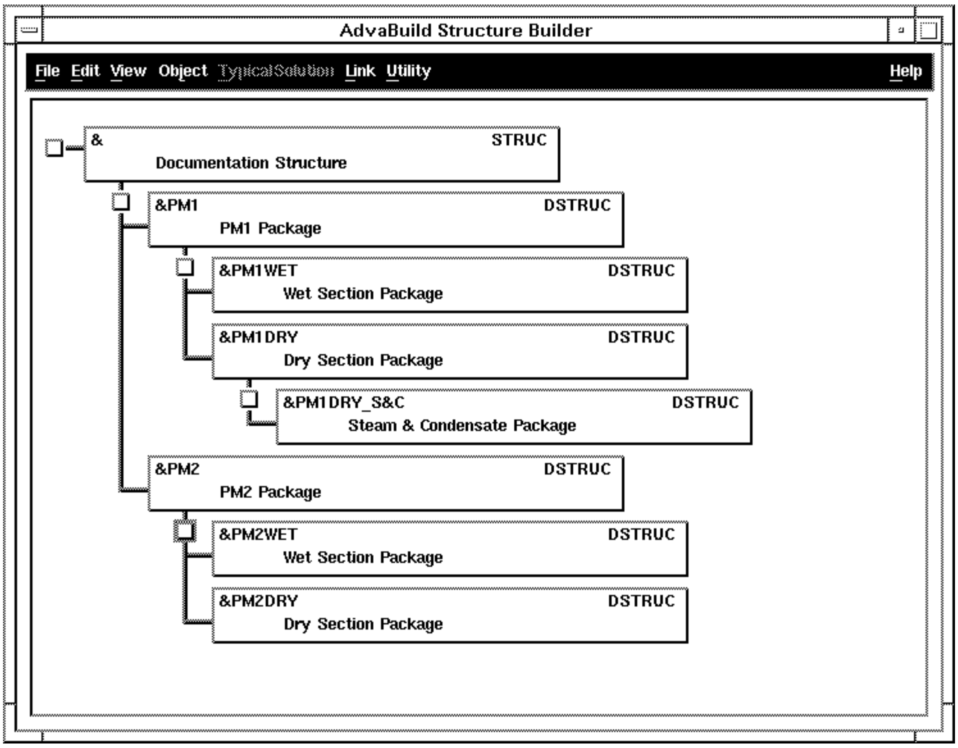


Figure 4-14. Plant Documentation Structure

4.1.5.1 Object-types in the Plant Documentation Structure

The plant documentation structure is built with the following AdvaBuild object-type:

Table 4-16. Objects in the Plant Documentation Structure

Object-type	Associated objects	Max. number of levels
DSTRUC	Documentation package	4

Figure 4-15 shows an example of a structure path and document package designation:

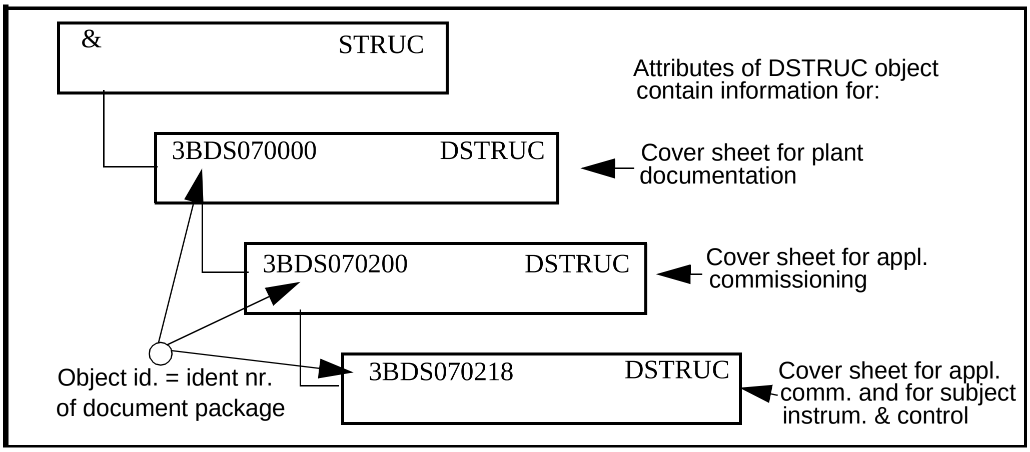

&
STRUC
3BDS070000
DSTRUC
3BDS070200
DSTRUC
3BDS070218
DSTRUC
Attributes of DSTRUC object contain information for:
Cover sheet for plant documentation
Cover sheet for appl. commissioning
Cover sheet for appl. comm. and for subject instrum. & control
Object id. = ident nr. of document package

Figure 4-15. Designation of Document Packages

The data for the coversheet of the plant documentation are assigned as attributes to the top structure-object, whereas the information for the coversheets of the relevant documentation packages are assigned as attributes on the subordinated documentation structure-objects.

4.1.5.2 Link-types to Objects in the Plant Documentation Structure

Objects in the plant documentation structure can be linked to objects in the function-, location-, and control system structures by the following link-type:

Table 4-17. Links to Objects in the Plant Documentation Structure

Link-type	Linked object-type	Example
IS_DOCUMENTATION_OF	DSTRUC to any object in the function-, location- and control system structure	Link from DSTRUC-object representing the documents for appl. “installation” to TSTRUC-object

When output for documentation is requested, the documentation of objects in the subtrees to which it was linked will be output. The output documents correspond to the predefined document-types of the documentation package(s).

4.1.5.3 Designation of Objects in the Plant Documentation Structure

The documentation-objects in the documentation structure are assigned an object identification number which is also used as identification number for the corresponding document package. This identification number has to be selected according to the ABB Industrial Systems Standard 3BSE000004.

4.2 List and Tree Presentation of Objects

The following object presentation is recommended:

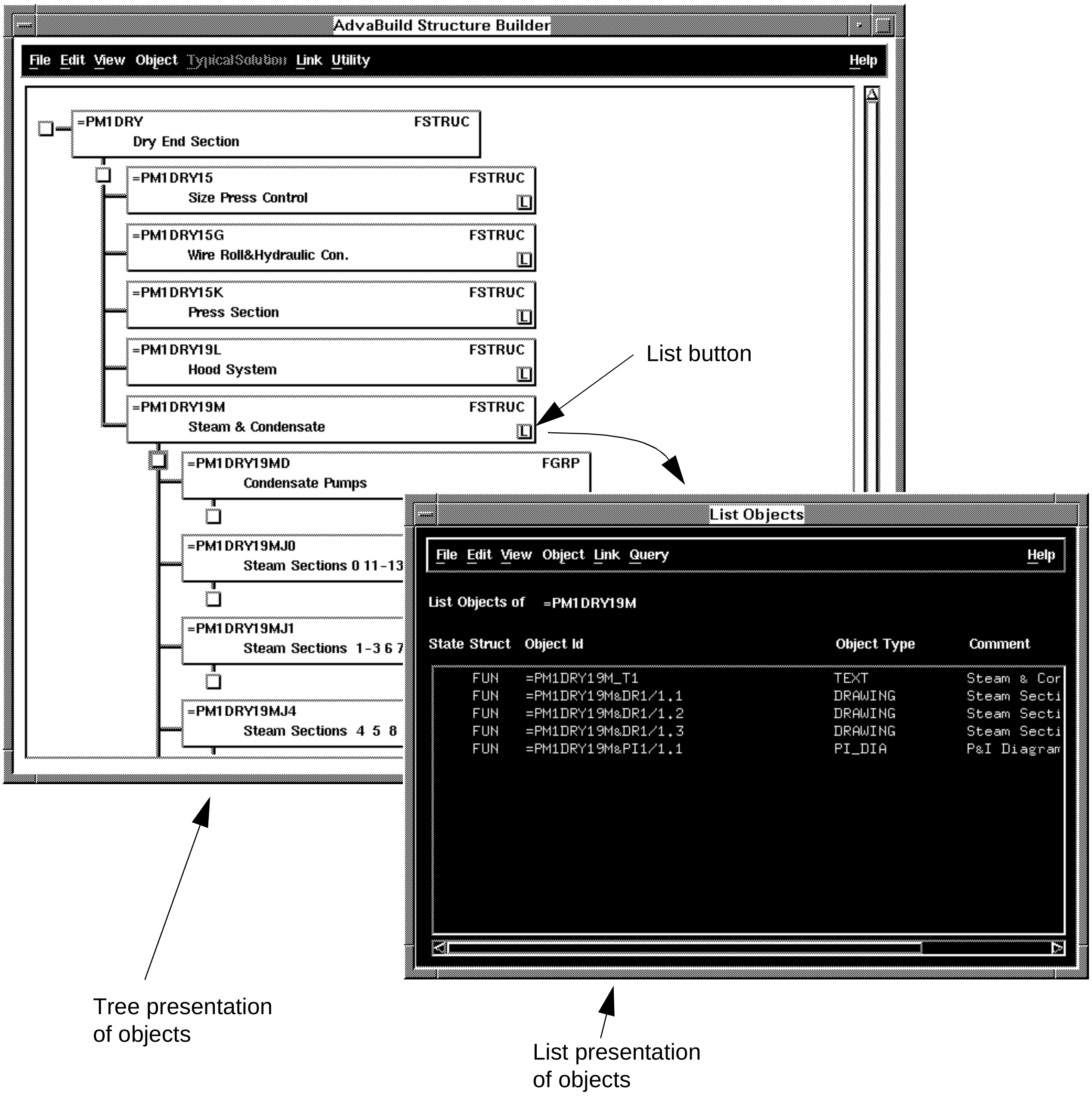
All object-types that do not have children objects should be presented in the list window of the Structure Builder. Such object-types represent:

- documents
- signal elements
- board elements
- device functions
- devices
- DCS-devices.

Objects which represent structure objects for the functional plant structure, the topographic location structure, the mounting location structure, and the control system structure should be presented in the tree window of the Structure Builder. Such object-types are:

- FSTRUC, FGRP, TAG
- TSTRUC, CUNIT, RACK
- control system objects down to the level 'node'.

Figure 4-16 shows examples for list and tree presentations of objects:



```
List Objects of   =PM1DRY19M

State Struct  Object Id                                Object Type    Comment

      FUN     =PM1DRY19M_T1                            TEXT           Steam & Cor
      FUN     =PM1DRY19M&DR1/1.1                       DRAWING        Steam Secti
      FUN     =PM1DRY19M&DR1/1.2                       DRAWING        Steam Secti
      FUN     =PM1DRY19M&DR1/1.3                       DRAWING        Steam Secti
      FUN     =PM1DRY19M&PI1/1.1                       PI_DIA         P&I Diagram
```

List button

Tree presentation
of objects

List presentation
of objects

Figure 4-16. List and Tree presentation of Objects

4.3 Creating Text, Drawing, and Diagram Objects

Text descriptions, diagrams, and drawings are represented by the following object-types in the project structure:

TEXT is an object type that contains text information and simple graphics. It is handled by the Document Builder.

DRAWING is an object type that contains free styled drawings. It is handled by the Diagram Builder.

PI_DIA is an object type that contains a pipe and instrumentation diagram. It is handled by the Diagram Builder.

HOOKUP is an object type that contains an assembly drawing for instrumentation. It is handled by the Diagram Builder.

CI_DIA is an object type that contains a circuit diagram. It is handled by the Electrical Diagram Builder.

LA_DIA is an object type that contains a layout diagram. It is handled by the Electrical Diagram Builder.

4.3.1 Creating and Editing TEXT-objects

TEXT objects are assigned to project objects, which represent that part of the plant which is dealt with by the TEXT object.

Inserting a TEXT-object and editing the information (text, data).	
Preconditions:	• Parent-object created to which TEXT-object can be assigned.
User activities:	• Use the Structure Builder. • Click on the parent object to which TEXT object shall be assigned. • Choose **Insert** from the **Edit** menu. • Select object-type TEXT, designate it uniquely, and comment it. • Insert TEXT-object. • Open TEXT-object, select view Template Builder, and create the document (default file name). • Open TEXT-object, select view Document Builder. • Edit text. • Exit Document Builder.
Result:	• TEXT-object inserted and edited.

According to the above procedure, the TEXT-object is created for presentation in the structure tree. Should the document be created for list presentation where it is hidden in the structure tree, the following procedure can be applied:

1. Use the Structure Builder.
2. Click on the parent object to which a TEXT-object shall be assigned.
3. Choose **Open Type** from the **Object** menu.
4. Select TEXT.
5. Template Builder window appears.
6. Insert TEXT-object by using the Template Builder (default file name).
7. Open TEXT-object, select view Document Builder.
8. Edit text.
9. Exit Document Builder.

4.4 Importing Text and Drawing/diagram Objects

4.4.1 Importing TEXT-objects

When TEXT-objects are opened in the Document Builder view the first time, a default file name for the text file is created and assigned as an attribute to the TEXT-object. All text files of a project are stored in a predefined directory. An externally created text file can be imported and copied to a TEXT-object. The user can edit the imported text file by opening the relevant TEXT-object.

Importing text.	
Preconditions:	• Externally created text file available on DOS diskette. • Parent object created to which text object can be assigned.
User activities:	• Import externally created file from DOS diskette to a predefined directory on the workstation by using HP-UX commands. • Insert TEXT-object by using the Structure Builder (for tree presentation) or by using the Template Builder (for list presentation). • Open TEXT-object, select view Template Builder to create the text file (default file name). • Open TEXT-object and select view Document Builder. • Choose **Import** from the **File** menu and select the text file to be inserted. – alternatively •Copy externally created text file to (empty) file of TEXT-object by using HP-UX commands. •Open TEXT-object to edit imported text by using the Document Builder. • Exit Document Builder.
Result:	• External created TEXT-object inserted.

4.4.2 Importing Drawing/diagram Objects

Importing and creating of P&I-diagrams (The same procedure is valid for all other diagram/drawing objects.)	
Preconditions:	• Externally created P&I-diagram file available on DOS diskette. • Parent object created to which P&I-diagram can be assigned.
User activities:	• Import externally created diagram file from DOS diskette to a predefined directory on the workstation by using HP-UX commands. • Insert PI_DIA-object by using the Structure Builder (for tree presentation) or by using the Template Builder (for list presentation). • Open PI_DIA-object, select view Template Builder to create the text file (default file name). • Open PI_DIA-object and select view Diagram Builder. • Import P&I-diagram file to file of PI_DIA-object with the import facility of the Diagram Builder. • Edit P&I-diagram by using the Diagram Builder. • Exit the Diagram Builder.
Result:	• P&I-diagram object inserted.

4.5 Solutions

4.5.1 Definition of Solutions

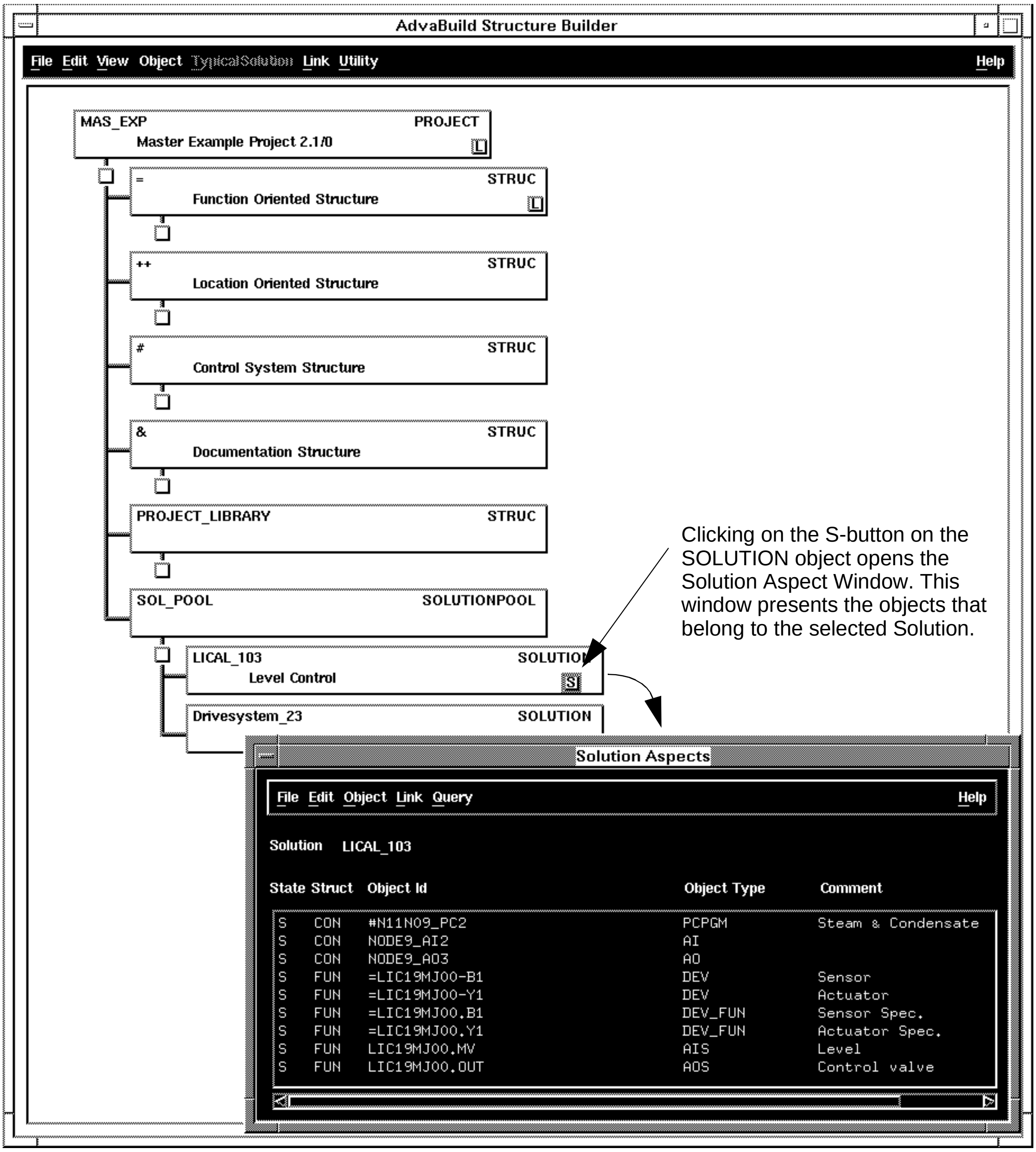
An additional structuring concept is provided by the use of Solutions. They allow you to group objects into units that have a certain significance within the engineering process, such as fulfilling a certain function or occupying a certain location. Such a group of objects is called a **Solution.**

Objects belonging to a Solution are called **Solution Aspect Objects**. The solution itself is represented by a unique object of the type SOLUTION. SOLUTION objects reside under a SOLUTIONPOOL object, which is a child of the PROJECT object. There can be any number of SOLUTIONPOOL objects under a PROJECT object, and any number of SOLUTION objects under a SOLUTIONPOOL object.

Creating a Solution object.	
Preconditions:	• Solutionpool created.
User activities:	• Use the Structure Builder. • Click on the SOLUTIONPOOL object. • Choose **Insert** from the **Edit** menu. • Select object-type SOLUTION, designate it uniquely, and comment it. • Insert SOLUTION object. • Open SOLUTION object, select view Template Builder, and fill in the attributes that describe the solution. • Press **Accept/Save** and exit the Template Builder.
Result:	• SOLUTION object created.

4.5.2 Content of Solutions

An object becomes part of a solution (Solution Aspect Object) by inserting it into the Solution Aspects window or by linking it to the Solution. The Solution Aspects window provides an overview and direct access to all Aspect Objects that belong to a Solution.



Clicking on the S-button on the SOLUTION object opens the Solution Aspect Window. This window presents the objects that belong to the selected Solution.

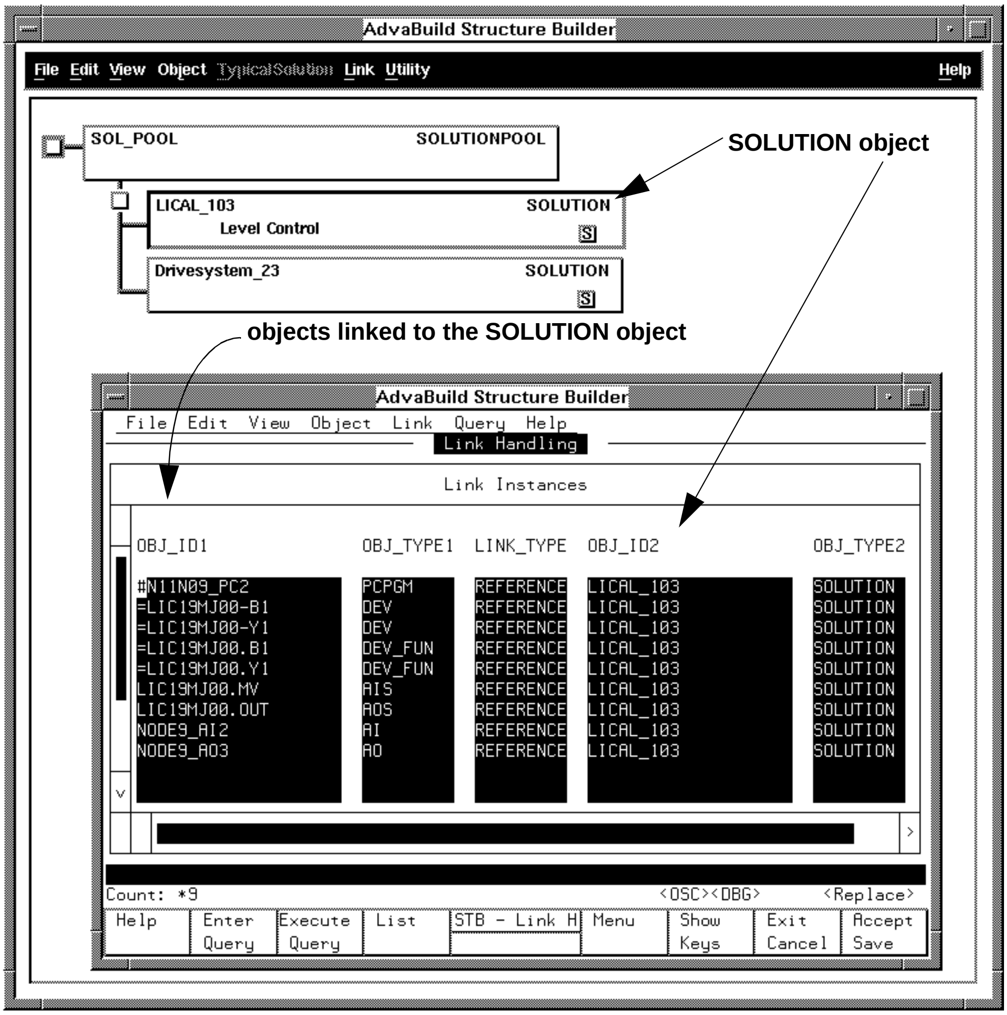
```
Solution    LICAL_103

State Struct  Object Id                         Object Type     Comment

S     CON     #N11N09_PC2                        PCPGM           Steam & Condensate
S     CON     NODE9_AI2                          AI
S     CON     NODE9_AO3                          AO
S     FUN     =LIC19MJ00-B1                      DEV             Sensor
S     FUN     =LIC19MJ00-Y1                      DEV             Actuator
S     FUN     =LIC19MJ00.B1                      DEV_FUN         Sensor Spec.
S     FUN     =LIC19MJ00.Y1                      DEV_FUN         Actuator Spec.
S     FUN     LIC19MJ00.MV                       AIS             Level
S     FUN     LIC19MJ00.OUT                      AOS             Control valve
```

Figure 4-17. Solution Object and Solution Aspects Window

Objects belonging to a Solution are linked to that Solution by the linktype REFERENCES_SOLUTION. Such objects represent a certain aspect of the overordinated Solution.



OBJ_ID1	OBJ_TYPE1	LINK_TYPE	OBJ_ID2	OBJ_TYPE2
#N11N09_PC2	PCPGM	REFERENCE	LICAL_103	SOLUTION
=LIC19MJ00-B1	DEV	REFERENCE	LICAL_103	SOLUTION
=LIC19MJ00-Y1	DEV	REFERENCE	LICAL_103	SOLUTION
=LIC19MJ00.B1	DEV_FUN	REFERENCE	LICAL_103	SOLUTION
=LIC19MJ00.Y1	DEV_FUN	REFERENCE	LICAL_103	SOLUTION
LIC19MJ00.MV	AIS	REFERENCE	LICAL_103	SOLUTION
LIC19MJ00.OUT	AOS	REFERENCE	LICAL_103	SOLUTION
NODE9_AI2	AI	REFERENCE	LICAL_103	SOLUTION
NODE9_AO3	AO	REFERENCE	LICAL_103	SOLUTION

Figure 4-18. Links of Objects to Solutions

Assign objects to solutions.

Preconditions:

- Solutionpool created
- Solution created

User activities for method 1 (object does not yet exist):

- Click on solution button of SOLUTION object.
- Choose **Insert** from the **Edit** menu in the solution Aspects window.
- Select object-type, designate it uniquely, and comment it. Fill in parent object to which selected object shall be assigned.
- Insert object.

User activities for method 2 (object exists):

- Use the Structure Builder.
- Click on the SOLUTION object.
- Choose **of Object** from the **Link** menu.
- Select linktype **REFERENCES_SOLUTION.**
- Link instance window appears.
- Fill in obj. id and object types of existing objects that shall be assigned to the Solution.
- Press **Accept/Save** and exit the Link Instance window.

Result:

- Object(s) assigned to SOLUTIONS.

4.6 Multi-user Engineering

General rules

A user can work on several projects.

Several users can work on a project.

Multi-user Features of the Builders

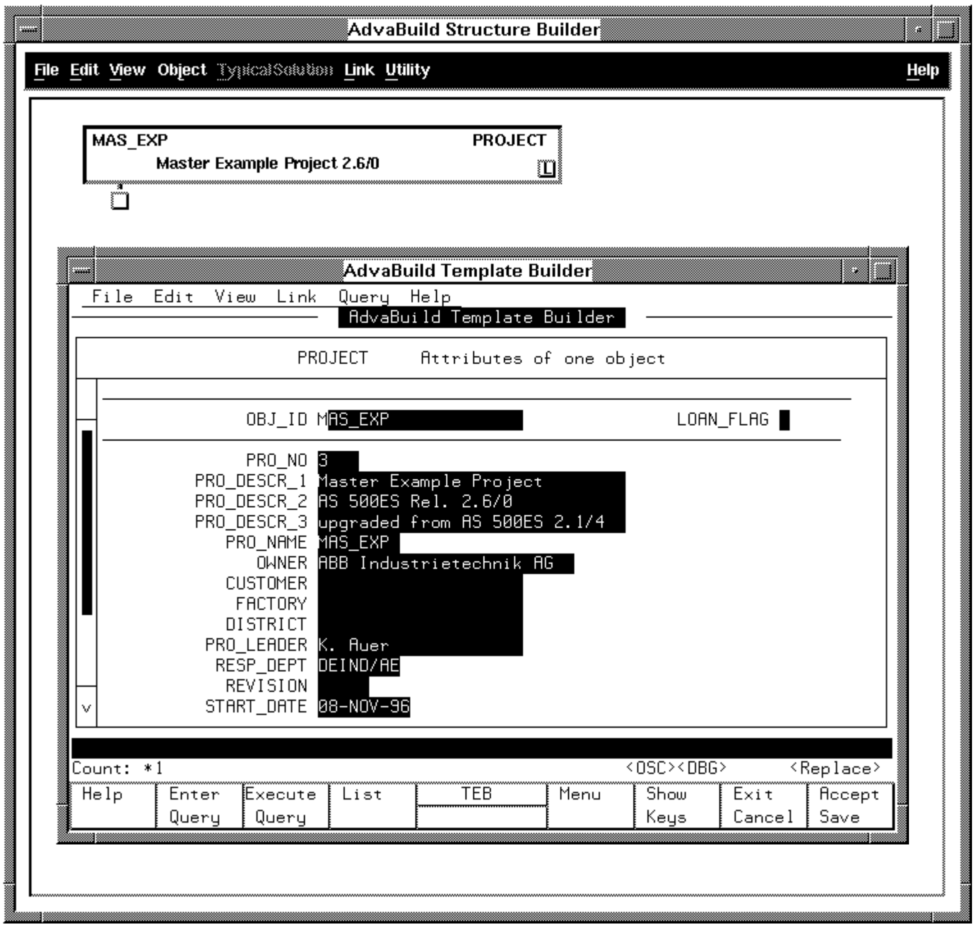
- Structure Builder:
  - A selected object will be locked for data input and transactions on it by further users after a transaction command is given or the last dialog "OK" for that transaction is entered.
  - A locked object will be unlocked after the transaction on the object is completed (for example, after executing the commands **Delete**, **Move**, **Copy**, and so on).
  - The data of the locked objects which were in the database before the locks were set will be displayed to a user for whom objects are locked.
- Template Builder:
  - An object is locked for manipulations by further users after an object attribute is modified (even if existing data are input again) or a transaction is initiated.
  - A locked object will be unlocked if the command **Save** is performed.
  - The data of the locked objects which were in the database before the locks were set will be displayed to a user for whom objects are locked.
- Document Builder and Diagram Builder:
  - When a document is opened by the Document Builder or Diagram Builder, it is locked for writing by other users. Another user can get access to the document and can read and edit it, but cannot save it to the database.
  - The document-object is unlocked after exit of the Document or Diagram Builder.
- Function Chart Builder:
  - Only one user can work on one node.
  - If database elements (via type-circuits) or links to database elements (connections) are inserted, the node object will be locked. The lock will be released after a 'save' command by the PC-Editor.
  - During code generation, backtranslation, and document generation, the node, the database elements linked to that node, and the links themselves will be locked.

Chapter 5 Basic Engineering

5.1 Opening a Project

A delivery project is defined by creating the root-object of a project structure. This is performed by the system administrator, who also defines the access rights for users. Basic information about the project (such as project name, customer, responsible person, and so on) is assigned as attributes to the root-object of a project.

Figure 5-1 shows an example of a project root and its attributes as presented by the Structure Builder and the Template Builder.


AdvaBuild Structure Builder
File Edit View Object TypicalSolution Link Utility Help
AdvaBuild Template Builder
File Edit View Link Query Help
AdvaBuild Template Builder
Help
Enter Query
Execute Query
List
TEB
Menu
Show Keys
Exit Cancel
Accept Save

MAS_EXP PROJECT
Master Example Project 2.6/0

PROJECT Attributes of one object

OBJ_ID MAS_EXP
LOAN_FLAG

PRO_NO 3
PRO_DESCR_1 Master Example Project
PRO_DESCR_2 AS 500ES Rel. 2.6/0
PRO_DESCR_3 upgraded from AS 500ES 2.1/4
PRO_NAME MAS_EXP
OWNER ABB Industrietechnik AG
CUSTOMER
FACTORY
DISTRICT
PRO_LEADER K. Auer
RESP_DEPT DEIND/AE
REVISION
START_DATE 08-NOV-96

Count: *1 <OSC><DBG> <Replace>

Figure 5-1. Presentation of the Project Object

Logging into AdvaBuild for a certain project is possible only for users with access rights. When a user logs into AdvaBuild, the Structure Builder is activated and creates its default window with the basic menu items. By selecting the relevant menu items, the user can navigate within existing object structures and can create additional objects. Initially, there is only the root-object. After logging in, the user can append children objects to the root.

5.2 Collecting General Project Information

Before starting basic engineering, such information as environmental conditions, energy supply conditions, media supply conditions, and so on, must be collected and stored in TEXT-objects. The basic principles of plant structuring and plant designations should be described in another TEXT-object. Two of these documents are defined in the standard documentation of the Plant Documentation Builder by the document-type "List of general data" and the document-type "List of designations". Figure 5-2 shows the presentation of a TEXT-object by the Document Builder for editing.

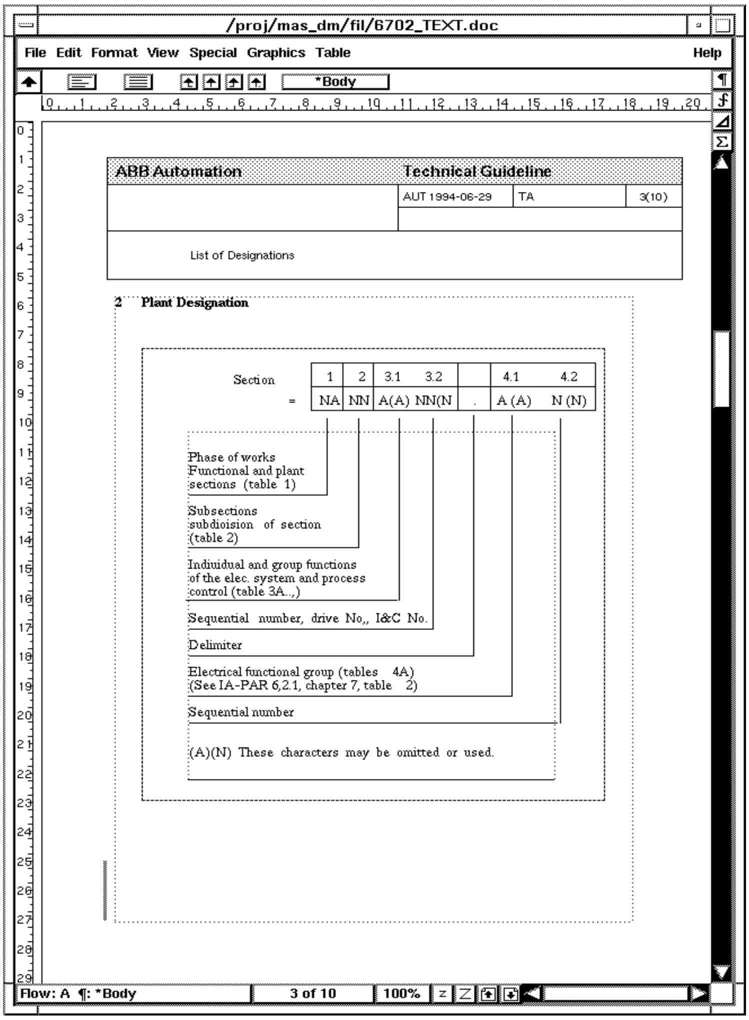


ABB Automation	Technical Guideline		
	AUT 1994-06-29	TA	3(10)
List of Designations			

2 Plant Designation

Section	1	2	3.1	3.2		4.1	4.2
=	NA	NN	A(A)	NN(N	.	A (A)	N (N)

Phase of works
Functional and plant
sections (table 1)

Subsections
subdioision of section
(table 2)

Indiuidual and group functions
of the elec. system and process
control (table 3A..,)

Sequential number, drive No,, I&C No.

Delimiter

Electrical functional group (tables 4A)
(See IA-PAR 6,2.1, chapter 7, table 2)

Sequential number

(A)(N) These characters may be omitted or used.

Figure 5-2. Presentation of a Text-Object

TEXT-objects are created to carry this information. The information is input either by using the edit functions of the Document Builder or by importing the information from an existing document by using the import function of the DOB.

5.3 Design of the Basic Plant Structures

5.3.1 Design of the Function Structure

The creation of this structure is started at the project-object. A STRUC-object which represents the functional view of the plant is inserted as a child-object. The FSTRUC-objects for the functional view of the subplants and, if necessary, further levels of FSTRUC-objects for parts of subplants are assigned as children below this STRUC-object. Tags which are defined in the P&I-diagrams are assigned to the FSTRUC-objects of the lower levels. Interacting tags may be assigned to functional groups, which are represented by FGRP-objects in a level between FSTRUC-objects and TAG-objects.

The plant structure can be redesigned at any time. When a structure-object is moved to another parent-object within the structure, the relationships of all children-objects and links are maintained. When a structure-object is deleted, the children-objects are also deleted.

5.3.2 Design of the Topographic-location Structure

The creation of this structure is started at the project-object. A STRUC-object which represents the topographic-location view of the plant is inserted as child to that node. TSTRUC-objects are added below this STRUC-object in a parent-child relationship. These TSTRUC objects are tree-like structures, with each object representing a topographic location. Objects of the mounting-location structure are assigned as children to the TSTRUC-objects of the lower levels.

The topographic-location structure can be redesigned at any time. When a structure-object is moved to another parent-object within the structure, the relationships of all children-objects and links of the structure object are maintained. When a structure-object is deleted, the children-objects are also deleted.

5.4 Creation/Selection of Project Standards

5.4.1 Type Circuits

Type Circuits are preconfigured and tested solutions for control functions and associated database elements. They are configured and edited by the configuration systems MA220 and MA720 and imported in AMPL source code to a predefined directory in the workstation.

The Type Circuits which should be used for a certain project are selected and imported to the workstation.

Import of Type Circuits from MA220 system to the workstation.	
Preconditions:	- Type Circuits created on configuration system MA220.
User activities:	- Select Type Circuits on MA220. - Generate sourcecode files on MA220 (.TCS). - Copy sourcecode files to virtual disc on workstation. - Convert sourcecode files and copy them to predefined directories in the workstation.
Result:	- Type Circuits available on predefined directory in workstation.

5.4.2 Item Designation

The application standard for the item designation system with regard to the higher levels of the plant structures has to be selected together with the customer or taken over from the customer. It is normally a text document which has to be imported and stored as a TEXT-object (see Section 5.2, Collecting General Project Information). It should be assigned as a child to the project node.

5.4.3 Man Machine Communication

These standards are only available as free-styled text and can be imported to a TEXT-object in the workstation, or this free-styled text will be created by the Document Builder.

The TEXT-object standards should be assigned as children to that STRUC-object which heads the function structure.

5.4.4 DCS Software Standards

Not supported, see Section 5.4.1, Type Circuits.

5.4.5 DCS Hardware Standards

Not supported.

5.4.6 Test Procedures

The test procedures are specified in text documents which can be imported to a TEXT-object in the workstation, or they can be specified by the Document Builder.

The TEXT-object for these standards should be assigned as children to that STRUC-object which heads the function structure.

5.4.7 Installation Standards

Installation standards comprise standards for:

- circuits (such as a standard circuit diagram for cabinets)
- arrangements (such as for cabinets)
- interconnections (such as for the interconnection of cabinets)
- grounding
- assemblies (such as hook-ups for field instrumentation).

These standards are assigned to document-objects (TEXT- and diagram-objects) within the basic project structures.

The STRUC-object which heads the topographic-location structure is the parent-object for the relevant text-objects and diagram-objects (TEXT-, LA_DIA-, CI_DIA-, DRAWING-objects) containing installation standards.

5.4.8 Import/Export Standards

Import/export standards are defined for the exchange of data between the workstation and external data repositories (such as between CAE-system of the customer/partner and AdvaBuild). The following must be specified:

- data source(s) and data destination(s)
- data carriers and data transmission facilities between source and destination
- data format.

5.4.9 Design Standards

The design standards are defined in text-documents which can be imported to a TEXT-object in the workstation, or they can be defined by the Document Builder.

The TEXT-object for these standards should be assigned as children to the STRUC-object which heads the function structure.

5.5 Basic Function Planning and Acquisition of Basic Engineering-data

5.5.1 Handling of P&I-diagrams

Measuring and actuating functions (tags) are defined in the P&I-diagrams. These diagrams contain the basic information for the creation of the TAG-objects.

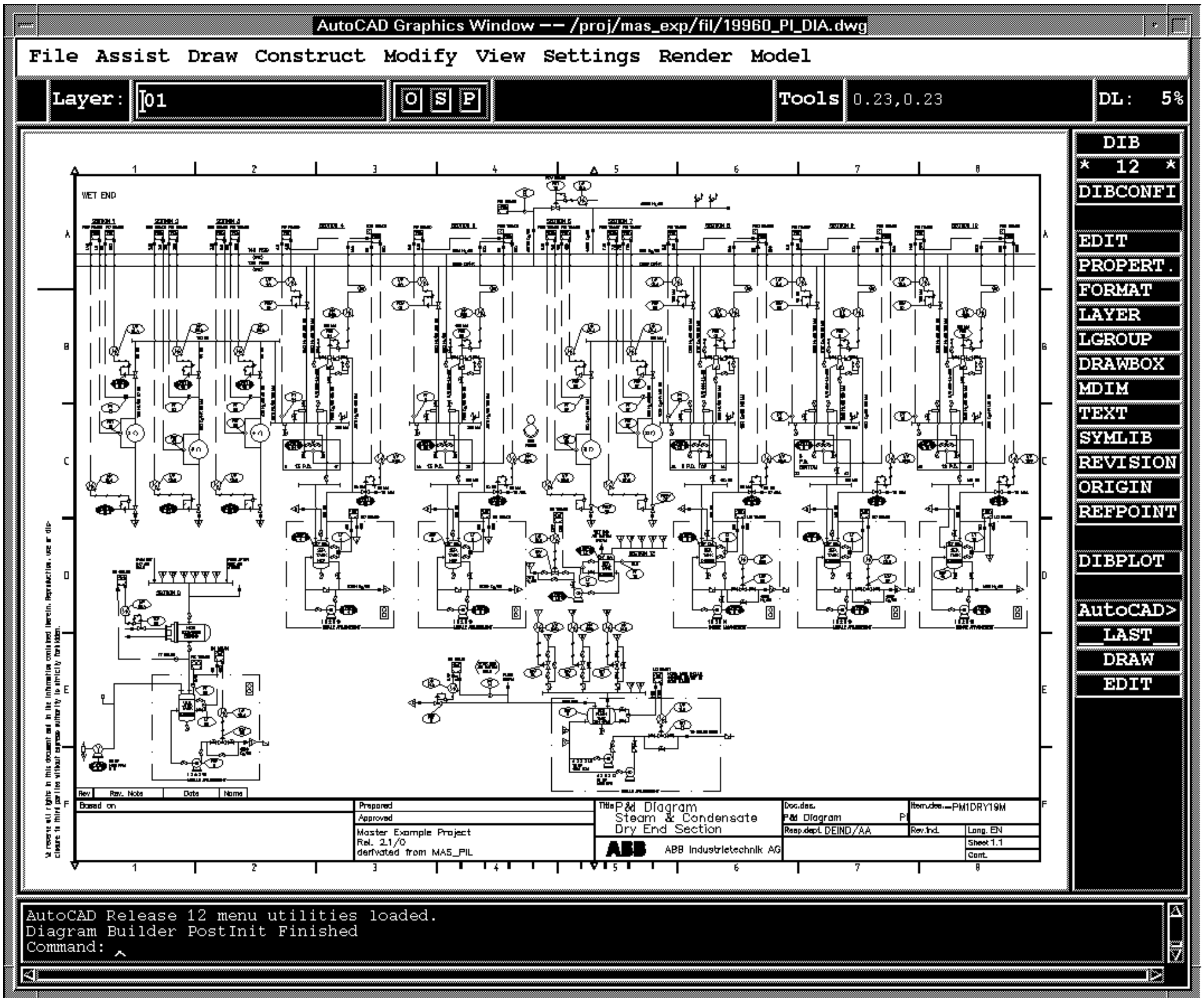
P&I-diagrams are based on process flow diagrams. They represent the technical realization of a process by means of graphical symbols for equipment and piping, together with graphical symbols for process measurement and control functions.

P&I-diagrams are handled by the Diagram Builder. In most cases, P&I-diagrams are created by the partner in the project who is responsible for the process. These external documents are imported to the workstation and assigned to PI_DIA-objects.

P&I-diagrams are accessed by opening the corresponding PI_DIA-objects, which are assigned as child-objects to FSTRUC-objects in the function-structure. After opening the object by using the Structure Builder, the Diagram Builder view is selected for the creation/alteration/import of P&I-diagrams.

A PI_DIA-object is assigned as child to the FSTRUC-object which represents the part of the plant which is covered by the P&I-diagram represented by the PI_DIA-object.

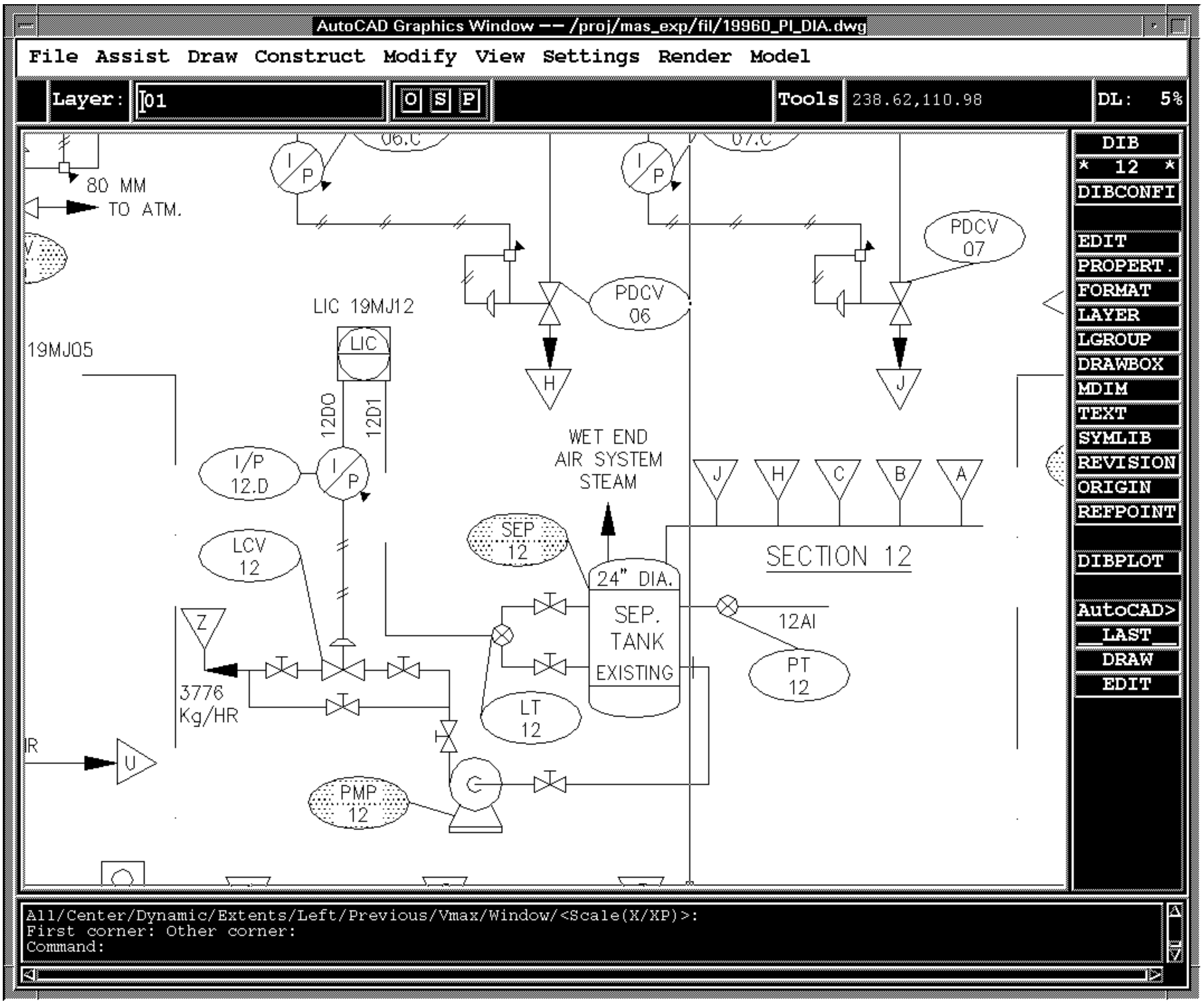
Figure 5-3 shows an example of a P&I-Diagram as presented by the Diagram Builder.



```
AutoCAD Release 12 menu utilities loaded.
Diagram Builder PostInit Finished
Command:
```

Figure 5-3. P&I-Diagram

Figure 5-4 shows an example for zooming into details of a P&I-diagram by using the Diagram Builder.



```
All/Center/Dynamic/Extents/Left/Previous/Vmax/Window/<Scale(X/XP)>:
First corner: Other corner:
Command:
```

Figure 5-4. Zooming into a P&I-Diagram

5.5.2 Assignment of Objects in the Topographic-location Structure to DangerAreas of the Plant

Danger areas in the plant have to be determined, and the classification of a danger area has to be assigned as an attribute to the objects of the topographic-location structure. This is done by opening the relevant structure-objects of the topographic-location structure and assigning the danger area code to the provided attribute with the Template Builder.

WARNING

Modifications in danger area classification may have severe consequences to already established planning results!

Editing an attribute of an object.

User activities:

- Open object and select view Template Builder.
- Edit attribute value by using Template Builder.
- Enter **Accept/Save** and exit Template Builder.

5.5.3 Specification of Man Machine Communication

The specification of basic principles of man machine communication is important for further activities of basic and detail engineering. This information has to be specified in a textual and/or free graphic form and is assigned to a TEXT-object or diagram-object, either with the Document Builder or with the Diagram Builder.

The access to these Builders is via TEXT- or diagram-objects, which are allocated to structure-objects in the function structure. No automatic data evaluation or transfer is available with these object-types.

The objects containing the MMC specification are assigned to FSTRUC-objects covering the plant section for which the MMC was specified.

5.5.4 Acquisition of Data for Measuring, Actuating, and Consuming Functions

In this step, the basic data for tags, DCS-signal elements, and device functions are collected. Compressed information sources for this data are:

- P&I-diagrams
- list of measuring points
- motor and component list
- specification sheet measurement and instrumentation.

With the exception of the P&I-diagrams (see Section 5.5.1, Handling of P&I-diagrams), these information sources are formatted text documents, which are often created by the customer or the partner responsible for process engineering. These documents are imported or created by the Document Builder as TEXT-objects.

Parent-objects for the TEXT-objects to which “list of measuring points”/“motor and components list” are assigned:

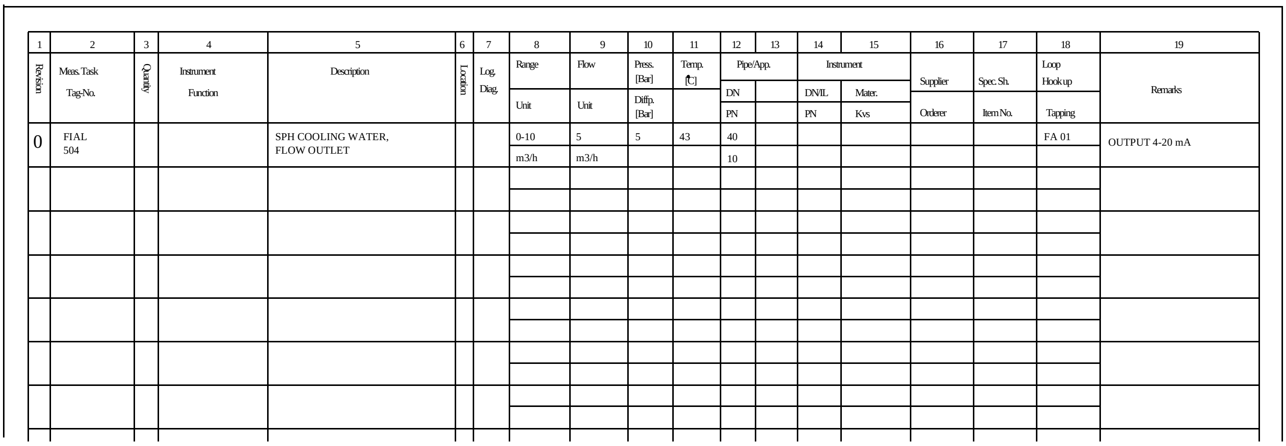
- STRUC-object which heads the function structure, if the list is not split up with regard to plant functions or sections,
- FSTRUC-objects if the list is split up with regard to plant functions or sections. It has to be assigned to those FSTRUC-objects which represent these plant functions or sections.

Parent-objects for the TEXT-objects to which the “specification sheets for measurement and instrumentation” are assigned:

- TAG-objects (the relevant TAG-objects may have been created in a previous step by evaluating the “list of measuring points”/“motor and component list”, or the P&I-diagrams).

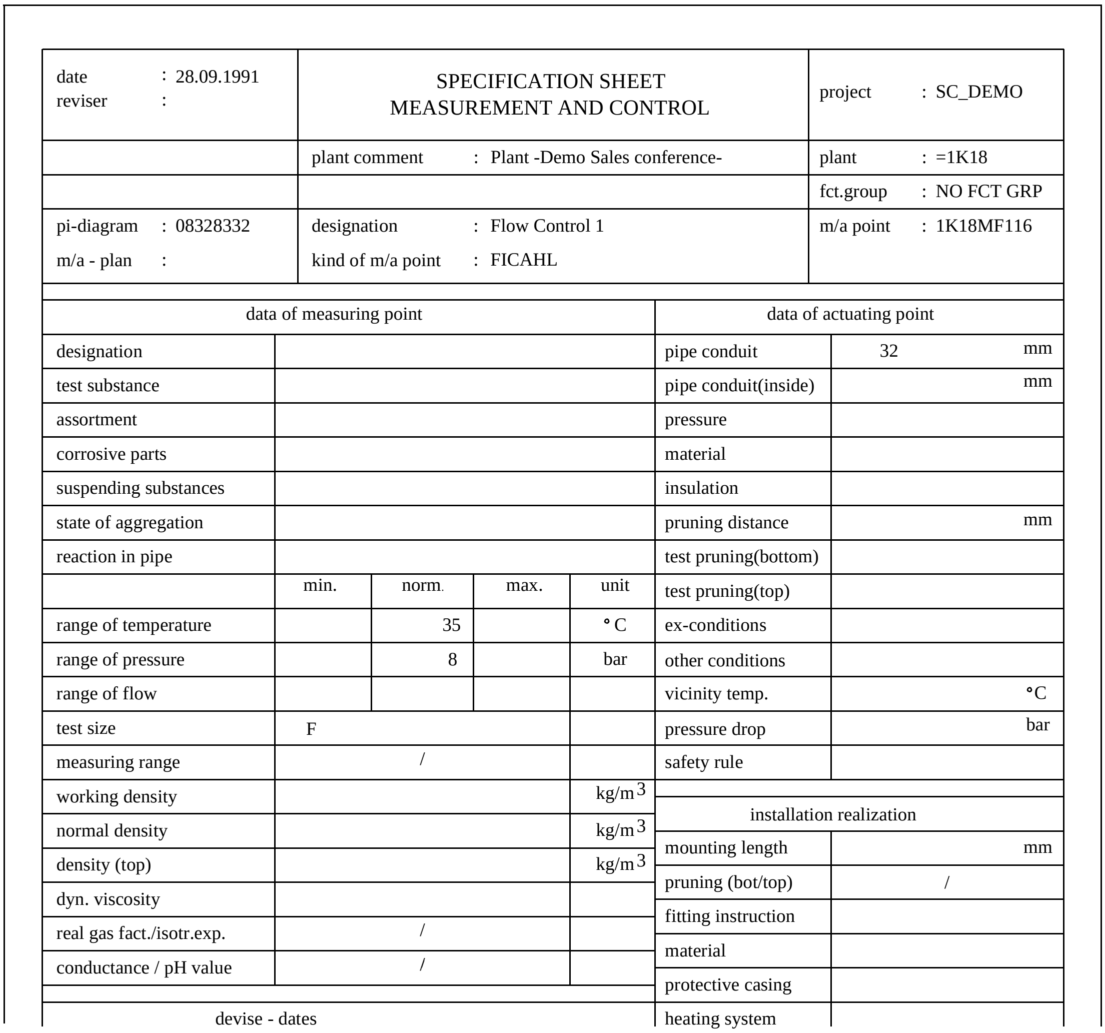
Figure 5-5 shows an example for a “List of Measuring Points”.

Figure 5-6 shows an example for a “Specification Sheet Measurement and Control”.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Revision	Meas. Task	Quantity	Instrument	Description	Location	Log. Diag.	Range	Flow	Press. [Bar]	Temp. [°C]	Pipe/App.		Instrument		Supplier	Spec. Sh.	Loop Hook up	Remarks
	Tag-No.		Function								DN		DN/IL	Mater.				
							Unit	Unit	Diffp. [Bar]		PN		PN	Kvs	Orderer	Item No.	Tapping	
0	FIAL 504			SPH COOLING WATER, FLOW OUTLET			0-10	5	5	43	40						FA 01	OUTPUT 4-20 mA
							m3/h	m3/h			10							

Figure 5-5. List of Measuring Points



date : 28.09.1991 reviser :	SPECIFICATION SHEET MEASUREMENT AND CONTROL	project : SC_DEMO
	plant comment : Plant -Demo Sales conference-	plant : =1K18
		fct.group : NO FCT GRP
pi-diagram : 08328332 m/a - plan :	designation : Flow Control 1 kind of m/a point : FICAHL	m/a point : 1K18MF116

data of measuring point					data of actuating point	
designation					pipe conduit	32 mm
test substance					pipe conduit(inside)	mm
assortment					pressure	
corrosive parts					material	
suspending substances					insulation	
state of aggregation					pruning distance	mm
reaction in pipe					test pruning(bottom)	
	min.	norm.	max.	unit	test pruning(top)	
range of temperature		35		°C	ex-conditions	
range of pressure		8		bar	other conditions	
range of flow					vicinity temp.	°C
test size	F				pressure drop	bar
measuring range	/				safety rule	
working density				kg/m^3		
normal density				kg/m^3	installation realization	
density (top)				kg/m^3	mounting length	mm
dyn. viscosity					pruning (bot/top)	/
real gas fact./isotr.exp.	/				fitting instruction	
conductance / pH value	/				material	
					protective casing	
devise - dates					heating system	

Figure 5-6. Specification Sheet Measurement and Control

Creation of TAG-objects, signal element objects, and device function-objects and input of data.

Preconditions:

- Process specified and basic function structure created
- P&I-diagrams created.

User activities:

- Insert the TEXT-objects for “list of measuring points”/“motor and component list” by using the Structure Builder and Template Builder.
- Open TEXT-objects, select view Template Builder to create the text documents (default file name).
- Open TEXT-objects, select view Document Builder, and import externally created text files to the relevant TEXT-objects.
- Insert the TAG-objects by using the Structure Builder or
- Import TAG-objects by evaluation of the tag definition in the list of measuring points.
  - Create DATA FILE, CONTROL FILE, and SELECT FILE, and store them in the directory reserved for export/import by the export/import utility of the Template Builder. The DATA FILE may be a copy of the list of measuring points if the CONTROL FILE and SELECT FILE correspond to its format.
  - Click on any object, select view Template Builder.
  - Choose **Import** from the **File** menu and enter filenames into the dialog box.
  - Click on **Accept/Save.**

User activities:

- Insert TEXT-object for “specification sheets measurement and instrumentation” and edit or import it as described above for the “list of measuring points”. These TEXT-objects are assigned as children to TAG-objects.
- Insert/import DCS-signal element objects and assign them as children to TAG-objects. Insertion and import may be accomplished in steps similar to those described for tags, if these signals are defined in external files.
- Insert/import device functions as DEV_FUN-objects and assign them as children to TAG-objects. Insertion and import may be accomplished in steps similar to those described for tags, if these device function-objects are defined in external files.

Result:

- Tag-objects, signal element objects, and device function objects are created and available data are assigned to these objects.

5.5.5 Definition of Hierarchical Relations of Control Functions

Functions which run the process must be evaluated from documents that describe the process (delivered by customer or partner). The control functions to be implemented in the control system are derived from these process functions.

The control functions have to be put into hierarchical relationships and are designated on a preliminary basis. This is shown in document-objects which keep the structure- and designation information of the preliminary defined control function objects.

Such a document-object is assigned as child to the STRUC-object which heads the function structure, if the whole plant is covered by the specified control functions. If only a part of the plant is covered by the document object, it is assigned to the FSTRUC object which represents that part.

Figure 5-7 shows the principles of a control function hierarchy.

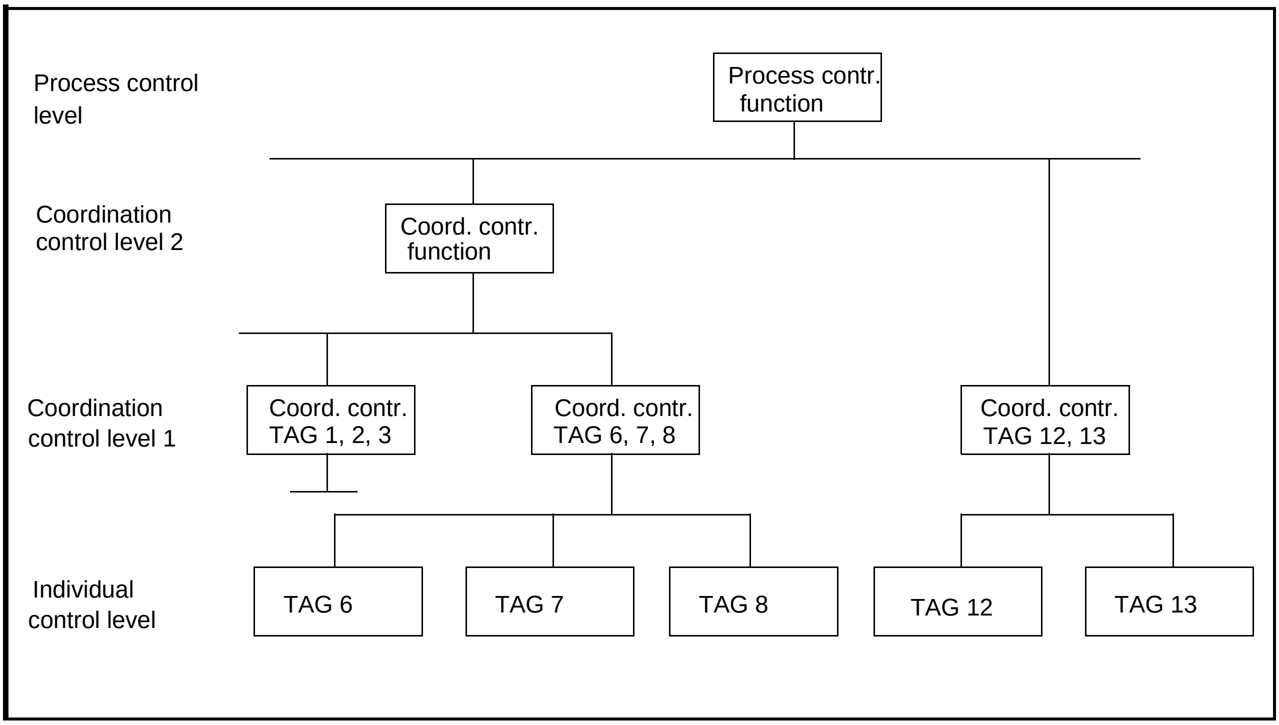


Figure 5-7. Control Function Hierarchy

5.5.6 Basic Specification of Control Functions and Assignment to Plant Structure Objects

The control functions which were defined, put into hierarchical relations, and designated on a preliminary basis, are basically specified one by one in separate TEXT- and/or diagram-objects, which are allocated as children-objects to objects of the function structure.

The function specifications may be created with the Document Builder and/or the Diagram Builder.

Parent-object for TEXT- or diagram-objects containing control function specifications:

- STRUC-object which heads the function structure, when the whole plant is covered by the specified control function,
- FSTRUC-object which covers the part of the plant for which a control structure is specified,
- FGRP-object for coordination functions for tags,
- TAG-object for specification of tag function.

Figure 5-8 shows an example for the specification of a control function in a function diagram:

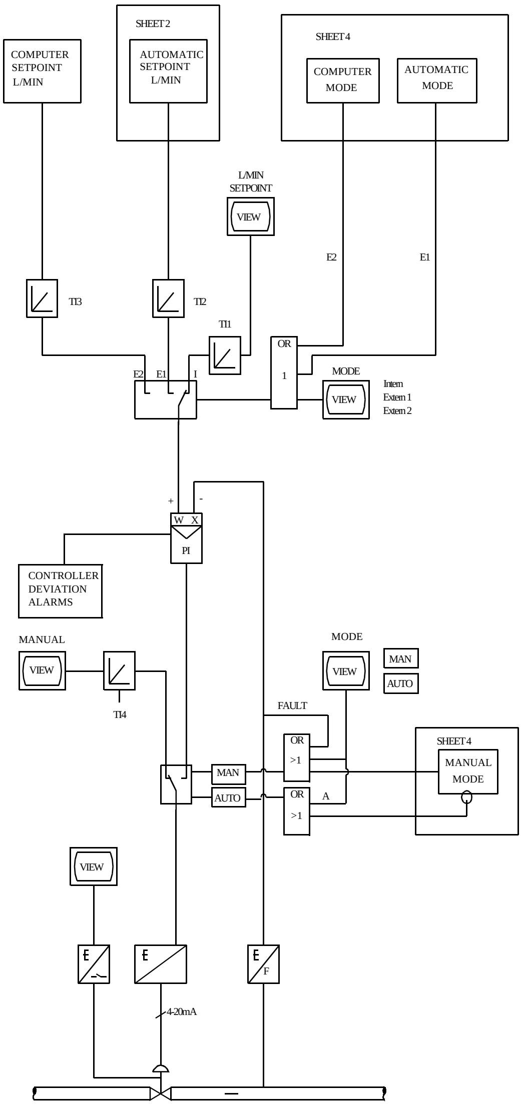


Figure 5-8. Function Diagram

5.6 Basic Configuration of the Control System

5.6.1 Design of the Control System Structure

The creation of the control system structure is based on the specifications according to the previous chapters regarding control functions and signal elements and based on the topographic location conditions.

The creation of this structure is started at the PROJECT-object. A STRUC-object which represents the control system view of the plant is assigned as a child to the PROJECT-object. The subtree below this STRUC-object is created by inserting the control system-objects.

Creation of a control system-object (such as MasterBus, MasterPiece, I/O-board) in the control system structure.	
Preconditions:	• Basic topographic-location structure created • Control functions basically specified • Amount of I/O signals estimated.
User activities:	• Use the Structure Builder. • Click on parent object to which control system object will be assigned. • Choose **Insert** from the **Edit** menu. • Select object-type to be inserted (such as MB300, MP200, PCPGM, PIDCON, AI, ...), designate it uniquely, and comment it. • Insert selected object.
Result:	• Object in the control system structure created and designated.

The control system structure can be redesigned at any time. When a structure-object is deleted, the children-objects are also deleted.

5.7 Basic Design of the Arrangement and Interconnection

5.7.1 Design of the Mounting-location Structure

The building up of mounting-location structures is started by creating and inserting the top structure-objects which represent mounting locations (MSTRUC-objects) or constructional mounting units (cabinets, desks, racks represented by the object-types CUNIT, SUBUNIT, RACK) and assigning them as children-objects to the topographic-location structure. If the mounting-location structure is created without knowing the proper assignment to the topographic-location structure (for example, a cabinet has to be defined and devices arranged before assigning it to a topographic location), a dummy object in the topographic-location structure has to be created to which all temporarily unassigned mounting-location structure trees are attached. The final assignment of an existing mounting-location structure to the topographic location will be accomplished by moving the top object of the mounting-location structure to its final destination object in the topographic-location structure (TSTRUC-object).

Creating and Moving a Mounting-location Structure

Click on a TSTRUC-object (it may be a dummy object, representing a pool of unassigned mounting-location objects or structure trees) via the Structure Builder. To that node add as a child-object either a CUNIT, SUBUNIT, or RACK-object which represents a constructional mounting unit. Below this top object continue building the mounting-location structure by inserting the relevant object-types. This mounting-location structure will later be moved to its final destination in the topographic-location structure:

Moving a mounting-location structure tree to a destination in the topographic-location structure.	
Preconditions:	• Mounting-location structure (at least top object of it) created • Basic topographic-location structure created.
User activities:	• Use the Structure Builder. • Click on top object of mounting-location tree which has to be moved. • Choose **Move** from the **Edit** menu. • Fill in the object id of the new parent-object and move the selected subtree.
Result:	• Mounting-location objects placed in topographic locations.

5.7.2 Acquisition of Basic Information for Arrangement and Interconnection

5.7.2.1 Basic Arrangement Planning

You must estimate where constructional units and what kind of constructional units will be placed in the plant on the basis of the topographical situation, the location of measuring, actuating, and consuming equipment, the location of MMC-control stations (see MMC-concept), and the structure of the control system. Possible topographic-location destinations for constructional units have to be selected.

This engineering step may result in a preliminary allocation of mounting-location objects (cabinets) to topographic-location objects and/or fixing it in the diagram-object "Overview Diagram Arrangement and Interconnection."

The arrangement planning of the constructional units is accomplished step by step. First, the constructional unit objects per type are created and assigned to the topographic-location-structure. The number of the assigned constructional units per type can be multiplied by using the menu item 'copy' of the Structure Builder (for example, five cubicles of one type may be assigned to one topographic location).

Existing constructional unit objects may be assigned to any topographic-location object in the plant with the menu item 'move' of the Structure Builder. Each constructional unit is assigned a unique designation (object id), independent of the designation of the topographic location.

To gain the information regarding not only the type but also the number of constructional units, the necessary devices to be located in constructional units may be allocated on a preliminary basis to constructional unit objects. The number of constructional units per type and their topographic location can be estimated from the number of devices according to the measuring, actuating, and consuming functions and the board-objects in the control system structure. Local refinements may be done during detail engineering (by inserting additional TSTRUC-objects, for example).

The allocation of devices to structure-objects in the subtrees of the constructional units and the precise number of constructional units will be determined during detail engineering.

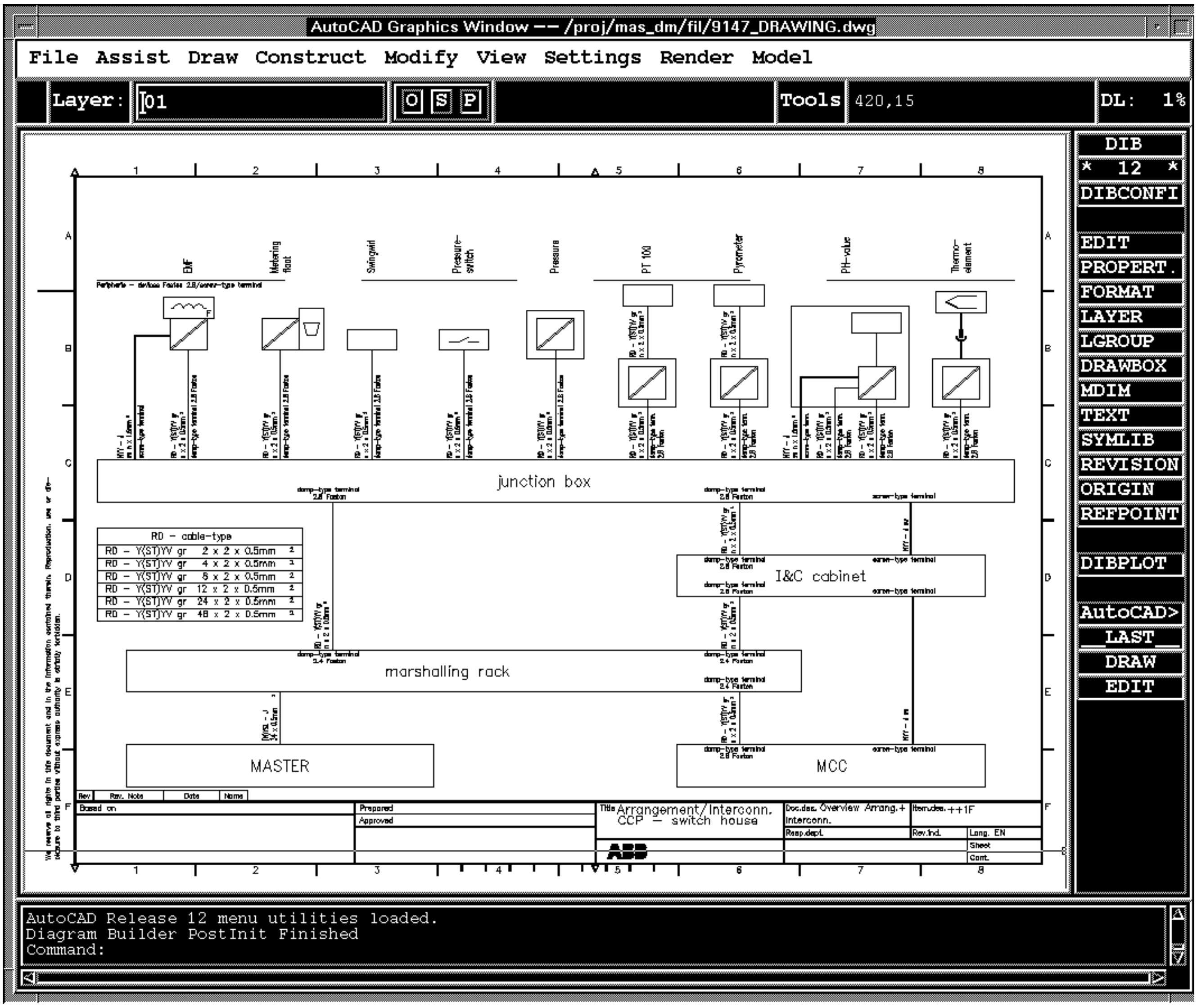
5.7.2.2 Specifications of Interconnections between Constructional Units (Objects of the Mounting-location Structure) and to Single Components

This engineering step is the basic design of the interconnections between the constructional units (such as cabinets and desks). The cable routes and the cable types are specified.

The selection of the cable routes is based on the document "Overview of Arrangement and Interconnection". The cable routes are specified in a document-object by the Diagram Builder or Document Builder as free diagrams or lists (document-type: Cable Routing Drawing/List).

In the present release of AdvaBuild, there is no tool which handles the data of this engineering step. However, it can be documented in the diagram-objects which give access to the files for the document-types "Overview of Arrangement and Interconnection" and "Cable Routing Drawing/List." These diagram-objects must be inserted below the top object of the topographic-location structure (STRUC-object).

Figure 5-9 shows an example of the document-type "Overview Diagram for Arrangement and Interconnection" as presented for editing by the Diagram Builder.



RD – cable-type
RD – Y(ST)YV gr 2 x 2 x 0.5mm ^{2}
RD – Y(ST)YV gr 4 x 2 x 0.5mm ^{2}
RD – Y(ST)YV gr 8 x 2 x 0.5mm ^{2}
RD – Y(ST)YV gr 12 x 2 x 0.5mm ^{2}
RD – Y(ST)YV gr 24 x 2 x 0.5mm ^{2}
RD – Y(ST)YV gr 48 x 2 x 0.5mm ^{2}

```
AutoCAD Release 12 menu utilities loaded.
Diagram Builder PostInit Finished
Command:
```

Figure 5-9. Overview Diagram for Arrangement and Interconnection

5.8 Planning of Documentation

Documentation is the output of engineering work. This output has to be planned, structured, assembled, administered, and presented in an appropriate manner. This is accomplished by the Plant Documentation Builder.

The Plant Documentation Builder is a shell function in the AdvaBuild environment for the collection of engineering documents produced by the various tools.

5.8.1 Planning of the Document-types within a Project

The Plant Documentation Builder provides lists of the standard document-types of plant engineering and their assignment to:

- applications

  such as documents for

  - erection
  - commissioning
  - plant operation
  - maintenance.
- subjects of delivery

  such as documents for

  - instrumentation and control
  - drive systems
  - energy supply systems.
- document classes

  such as documents for

  - process specifications
  - function specifications
  - arrangement specifications
  - wiring and interconnection
  - construction and assembly.

The documentation for a certain project can be planned individually. This is supported by the Plant Documentation Builder, which provides aids for the project related modification of the standard documentation and its above explained assignment.

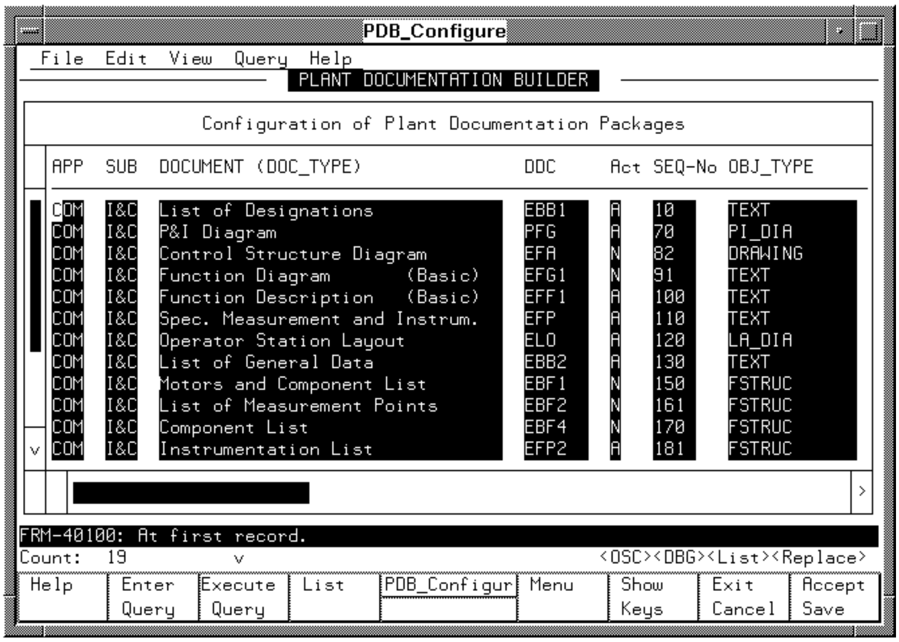
Out of the standard documentation are selected those document-types which are relevant for the project under consideration and which were agreed upon with the customer. There may also be the need to insert additional document-types for the project under consideration. This project related list of documents is the basis for the activation of printouts of documentation.

Display of predefined standard document-types.

User activities:

- Use the Structure Builder.
- Choose **Configure Plant Documentation Builder** from the **Utility** menu.
- Choose **Enter** from the **Query** menu in the configure template of the Plant Documentation Builder.
- Enter query conditions into attribute columns, use % as wildcard, leave columns empty for "all".
  Example: By entering "Basic Engineering" into column 'application' and by entering "Instrumentation and control" into column 'subject', all predefined documents will be displayed which might be applicable in the Basic Engineering of the subject Instrumentation and Control.
- Execute the query.

An example of the display of predefined standard documents by the Plant Documentation Builder is shown in Figure 5-10:



Configuration of Plant Documentation Packages

APP	SUB	DOCUMENT (DOC_TYPE)	DOC	Act	SEQ-No	OBJ_TYPE
COM	I&C	List of Designations	EBB1	A	10	TEXT
COM	I&C	P&I Diagram	PFG	A	70	PI_DIA
COM	I&C	Control Structure Diagram	EFA	N	82	DRAWING
COM	I&C	Function Diagram (Basic)	EFG1	N	91	TEXT
COM	I&C	Function Description (Basic)	EFF1	A	100	TEXT
COM	I&C	Spec. Measurement and Instrum.	EFP	A	110	TEXT
COM	I&C	Operator Station Layout	ELO	A	120	LA_DIA
COM	I&C	List of General Data	EBB2	A	130	TEXT
COM	I&C	Motors and Component List	EBF1	N	150	FSTRUC
COM	I&C	List of Measurement Points	EBF2	N	161	FSTRUC
COM	I&C	Component List	EBF4	N	170	FSTRUC
COM	I&C	Instrumentation List	EFP2	A	181	FSTRUC

```
FRM-40100: At first record.
Count:  19          v                         <OSC><DBG><List><Replace>
```

Figure 5-10. Plant Documentation Builder Configure Window

Delete a document-type within a certain context (such as Application, Subject).

User activities:

- Use the Structure Builder.
- Choose **Configure Plant Documentation Builder** from the **Utility** menu.
- Choose **Enter** from the **Query** menu in the configure template of the Plant Documentation Builder.
- Enter query conditions into attribute columns, use % as wildcard, leave columns empty for "all".
  Example: To delete Document-type "Motors and Component List" from all applications, enter "Motors and Component List" into column DOC_TYP.
- Execute the query.
- Select row in the display containing the document-type in its context which shall be deleted.
- Choose **Delete** from the **Edit** menu in the configure template of the Plant Documentation Builder.
- Click on **Accept/Save.**
- Repeat the deletion activity for each row in the display which shall be deleted (all rows in this example).

Insert a new document-type and its context.

User activities:

- Use the Structure Builder.
- Choose **Configure Plant Documentation Builder** from the **Utility** menu.
- Choose **Insert** from the **Edit** menu in the configure template of the Plant Documentation Builder.
- Enter attribute values into columns (use list functions to select valid values).
- Click on **Accept/Save.**

Result:

- Document-type with its context definition is part of the project documentation standard.

5.8.2 Creation of the Document Structure

Click on the project-object in the Structure Builder. To that node, add as a child-object a STRUC-object which represents the documentation view of the plant. Below this STRUC-object, assign in a parent-child relationship the DSTRUC-objects which represent documentation packages. The specification of the documentation package is done via the Template Builder. The specification data, which are attributes of the DSTRUC-objects, will be printed on the coversheets heading the documentation packages. DSTRUC-objects will be linked to objects (subtrees) of the function- and/or location- and/or control system structures. These links define the scope of the plant documentation.

Creation of a documentation package (DSTRUC-object) in the documentation structure.

User activities:

- Use the Structure Builder.
- Click on parent-object to which DSTRUC-object shall be assigned.
- Choose **Insert** from the **Edit** menu.
- Select object-type DSTRUC to be inserted, designate it uniquely, and comment it.
- Insert DSTRUC-object.

Result:

- Document package created.

Inserting attributes to objects in the documentation structure.

Preconditions:

- Documentation structure created.

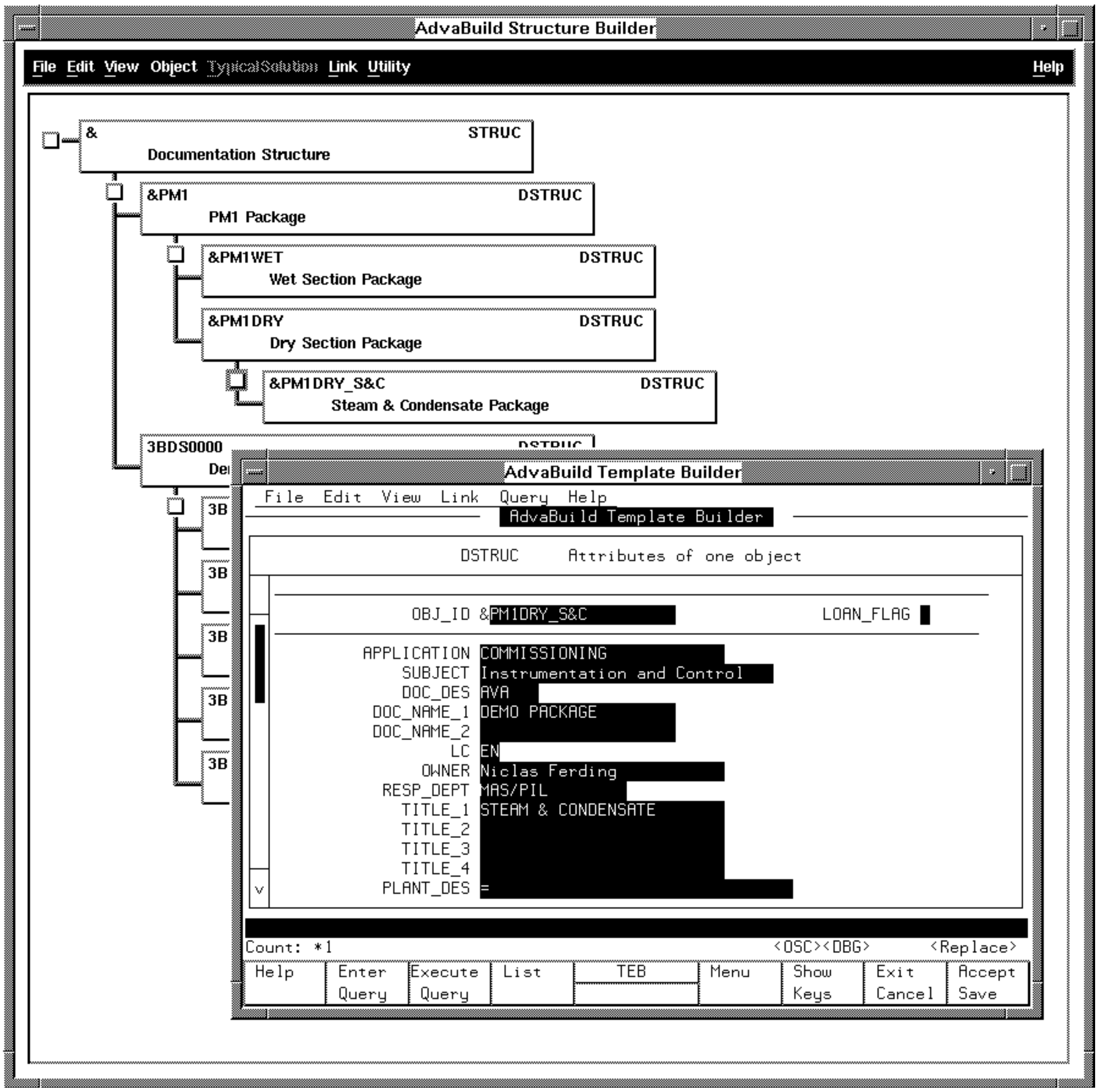
User activities:

- Use the Structure Builder.
- Open DSTRUC-object which shall be specified and select view Template Builder.
- Insert the attributes which specify the documentation package.
- Save the package specification data.

Result:

- Coversheet information of document package is specified.

Figure 5-11 shows the presentation of a Documentation Package by Structure Builder and by Template Builder.


AdvaBuild Structure Builder
File Edit View Object TypicalSolution Link Utility Help
AdvaBuild Template Builder
File Edit View Link Query Help
AdvaBuild Template Builder
Help
Enter Query
Execute Query
List
TEB
Menu
Show Keys
Exit Cancel
Accept Save

& STRUC
Documentation Structure

&PM1 DSTRUC
PM1 Package

&PM1WET DSTRUC
Wet Section Package

&PM1DRY DSTRUC
Dry Section Package

&PM1DRY_S&C DSTRUC
Steam & Condensate Package

3BDS0000

DSTRUC Attributes of one object

OBJ_ID &PM1DRY_S&C
LOAN_FLAG
APPLICATION COMMISSIONING
SUBJECT Instrumentation and Control
DOC_DES AVA
DOC_NAME_1 DEMO PACKAGE
DOC_NAME_2
LC EN
OWNER Niclas Ferding
RESP_DEPT MAS/PIL
TITLE_1 STEAM & CONDENSATE
TITLE_2
TITLE_3
TITLE_4
PLANT_DES =

Count: *1 <OSC><DBG> <Replace>

Figure 5-11. Structure Builder and Template Builder presentations of a Documentation Package

Creation of a link from a documentation package (DSTRUC-object) to objects in the function and/or location and/or control system structures.

Preconditions:

- Documentation structure and other relevant structures (function, location, control system) defined.

User activities:

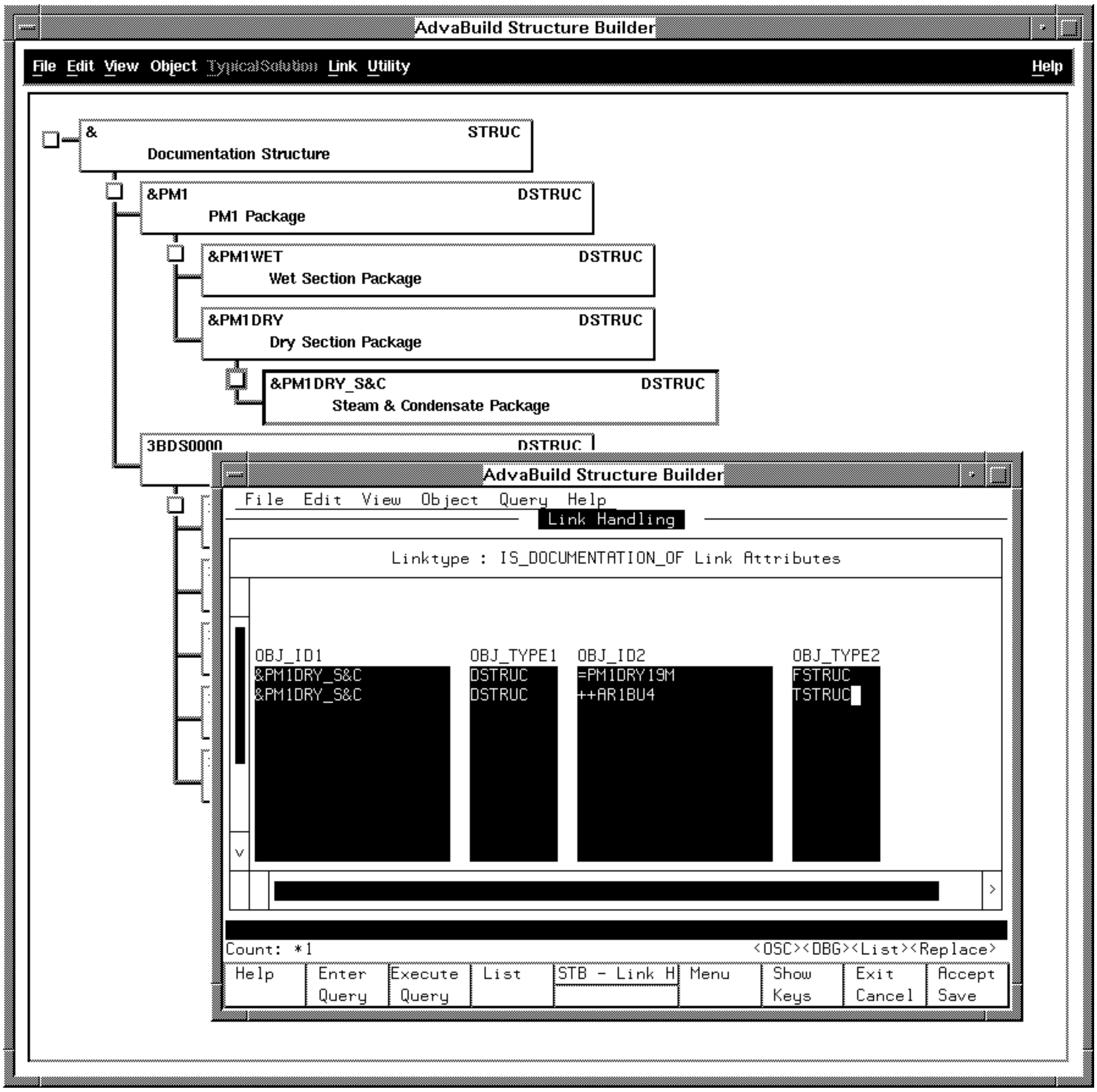
- Use the Structure Builder.
- Choose **Open Type** from the **Link** menu.
- Select link-type IS_DOCUMENTATION_OF.

  Link handling window is displayed.
- Choose **Insert** from the **Edit** menu.
- Input object identities of the objects to be linked.
- Click on **Accept/Save.**

Result:

- Documents in the subtree specified by the IS_DOCUMENTATION_OF link, and whose document type matches the types selected for the documentation package, are assigned to the documentation package.

Figure 5-12 shows the links of an object in the document structure:



Linktype : IS_DOCUMENTATION_OF Link Attributes

OBJ_ID1	OBJ_TYPE1	OBJ_ID2	OBJ_TYPE2
&PM1DRY_S&C	DSTRUC	=PM1DRY19M	FSTRUC
&PM1DRY_S&C	DSTRUC	++AR1BU4	TSTRUC

Figure 5-12. Links of a Documentation Package

Chapter 6 Detailed Engineering

6.1 Detailed Tag Engineering

6.1.1 Objects and Information in the Substructure of Tags

The tag functions are specified in the basic engineering phase by the following documents:

- P&I-diagram
- list of Measuring Points (Specification)
- specification Sheet Measurement and Instrumentation
- instrumentation List (Specification)
- function Diagram/Function Description.

Objects in the substructures of tags are created according to these specifications.

6.1.1.1 DCS-signal Elements

DCS-signal elements represent input and output signals of the control system. They are created as children of tags, but may be created alternatively in the control system structure. The signal elements are inserted by the Structure Builder or Template Builder. Their attributes are specified by the Template Builder.

Table 6-1 shows examples of DCS-signal element objects.

Table 6-1. DCS-Signal Element Objects

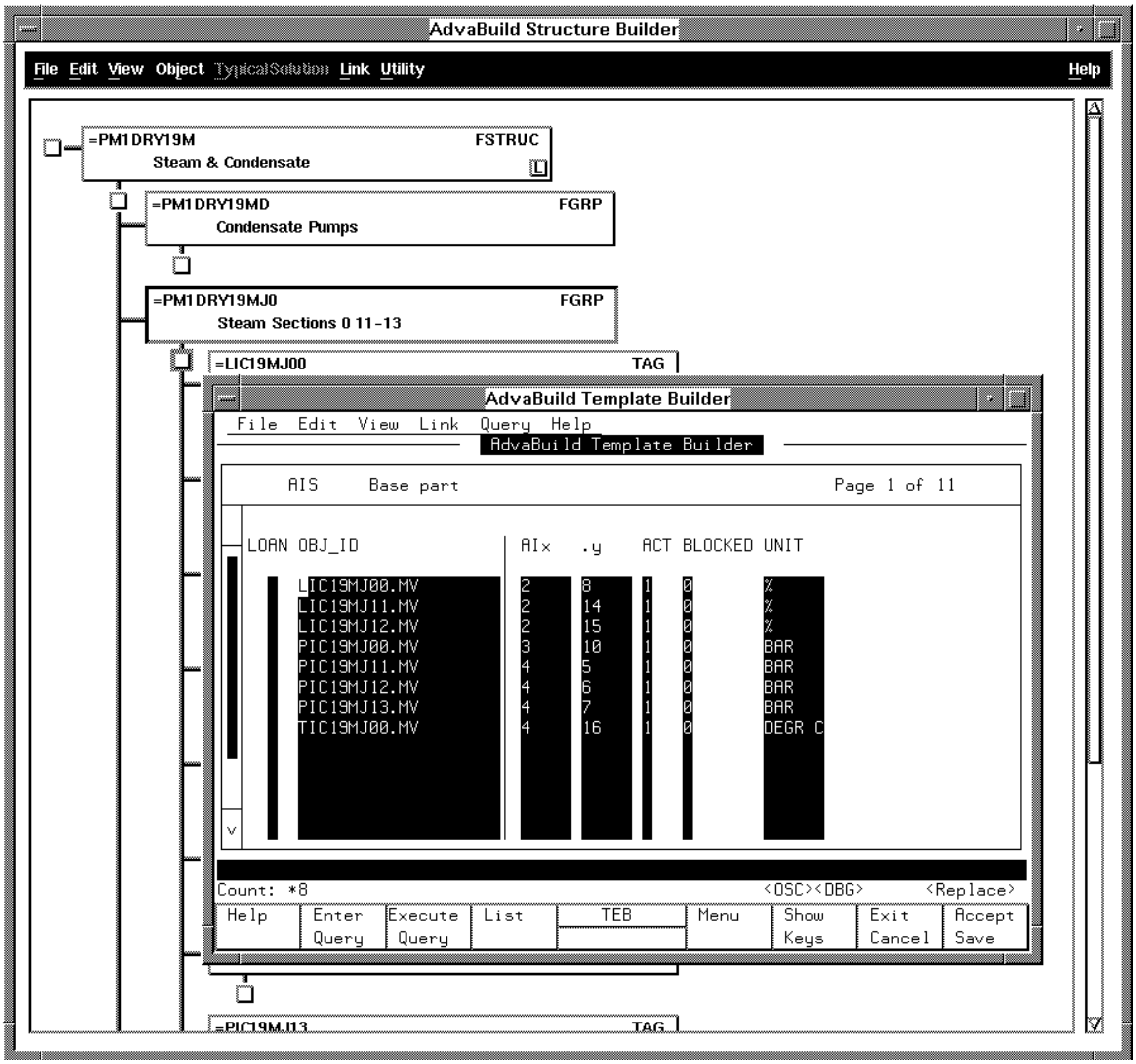
Object-type	Parent-objects	Associated signal elements
AIS	TAG, MA200	Analog Input
AOS	TAG, MA200	Analog Output
DIS	TAG, MA200	Digital Input
DOS	TAG, MA200	Digital Output

Note that there are far more DCS-signal elements defined for the MASTER system!

DCS-signal element objects are linked to board element objects in the control system structure by the link-type HAS_SIGNAL_ELEMENT. The links can be closed after the board element objects are inserted. (When a board element object is inserted into the control system structure, open links for the board channels are created automatically).

Instantiation of a DCS-signal (for list presentation) and inserting its attributes.	
Preconditions:	• Function structure with TAG-object created • Tag function specified (in "List of Measuring Points" and "Specification Sheet Measurement and Instrumentation," and, if necessary, in additional TEXT- and/or diagram-objects).
User activities:	• Use the Structure Builder. • Click on TAG-object to which signal element object shall be assigned. • Choose **Open Type** from the **Object** menu. • Select signal-object-type to be inserted from selectorbox (such as AIS). • Choose **Insert** from the **Edit** menu in the Template Builder window. • Click on **Accept/Save.**
Result:	• Signal element object created as child of a TAG-object and attribute values inserted.

Figure 6-1 shows the multi-element presentation of DCS signal element objects by the Template Builder:



AIS Base part Page 1 of 11

LOAN	OBJ_ID	AIx	.y	ACT	BLOCKED	UNIT
	LIC19MJ00.MV	2	8	1	0	%
	LIC19MJ11.MV	2	14	1	0	%
	LIC19MJ12.MV	2	15	1	0	%
	PIC19MJ00.MV	3	10	1	0	BAR
	PIC19MJ11.MV	4	5	1	0	BAR
	PIC19MJ12.MV	4	6	1	0	BAR
	PIC19MJ13.MV	4	7	1	0	BAR
	TIC19MJ00.MV	4	16	1	0	DEGR C

Count: *8 <OSC><DBG> <Replace>

Figure 6-1. Presentation of DCS Signal Element Objects by Template Builder

6.1.1.2 Device Function Objects and Device Objects

Device function objects represent the functional specification of peripheral devices (such as sensors, transmitters, and actuators) belonging to a tag.

Device objects represent a physical device that is used in the plant.

Device function objects and device objects are inserted by the Structure Builder.
The designation of a device object (object id) is used as item designation for the device.

Table 6-2. Device and Device-function Objects

Object-type	Parent-object	Associated device-objects
DEV_FUN	TAG, FGRP	Devices belonging to a TAG, except input/output devices of the control system.
DEV	TAG, FGRP, RACK, MSTRUC, TSTRUC	Physical Devices or Articles except input/output devices of the control system.

Creation of a device function or a device as child of a TAG object (for list presentation).

Preconditions:

- Function structure with TAG-object created
- Tag function specified (in "Specification Sheet Measurement and Instrumentation", for example)

User activities:

- Use the Structure Builder.
- Click on TAG-object.
- Chose **Open Type** from the **Object** menu.
- Select DEV_FUN object respective DEV object from selectorbox.
- Choose **Insert** from the **Edit** menu in the Template Builder window.
- Input object designation and attribute values.
- Click on **Accept/Save.**

Result:

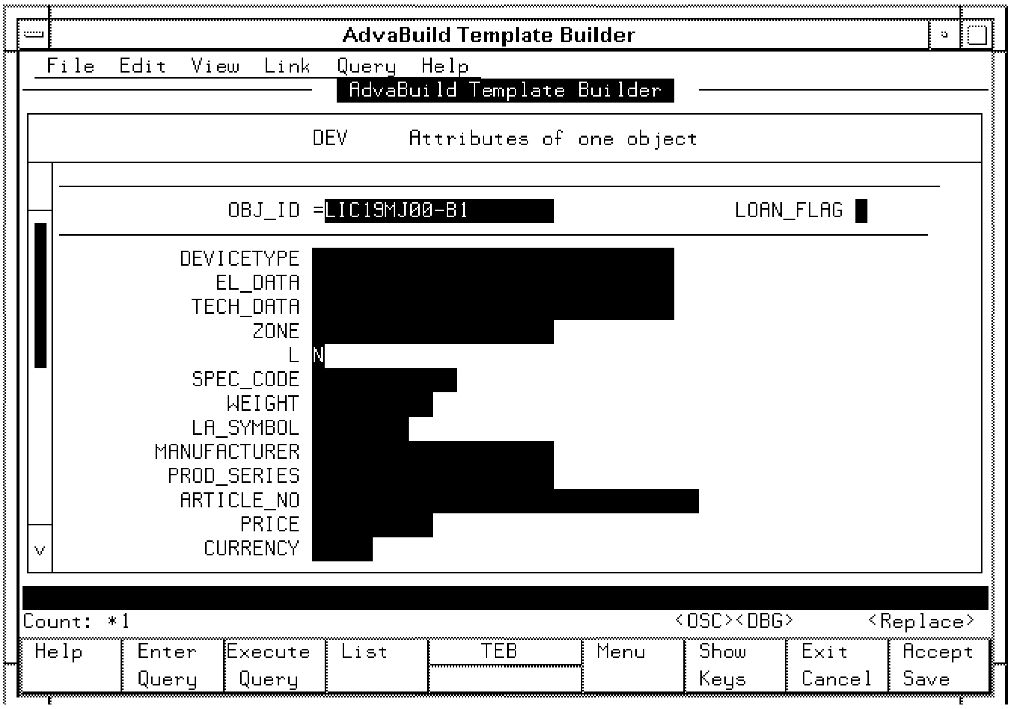
- Device function object respective device object created and attributes inserted.

The device function object respective the device object is inserted for display in list presentation according to the user activities listed above. This is the normal way device functions and devices should be presented.

Device function objects carry in attributes information about:

- related functions of tag
- zone of device allocation
- type designation of device
- name of producing company
- name of responsible person/company for ordering
- price relevant for calculation
- identifier for specification sheet.

The attributes of device objects are shown in Figure 6-2.


AdvaBuild Template Builder
File Edit View Link Query Help
AdvaBuild Template Builder
DEV Attributes of one object

OBJ_ID =LIC19MJ00-B1 LOAN_FLAG
DEVICETYPE
EL_DATA
TECH_DATA
ZONE
L N
SPEC_CODE
WEIGHT
LA_SYMBOL
MANUFACTURER
PROD_SERIES
ARTICLE_NO
PRICE
CURRENCY

Count: *1 <OSC><DBG> <Replace>

Help
Enter Query
Execute Query
List
TEB
Menu
Show Keys
Exit Cancel
Accept Save

Figure 6-2. Attributes of a Device Object

6.1.1.3 Document Objects

The following document objects may be assigned as children to TAG-objects. Further explanation with regard to the context of these documents in the engineering process will be given in the following sections.

Table 6-3. Documents Assigned to Tags

Document-type	Object-type	Builder used to create document	Parent-object
Circuit Diagram	CI_DIA	Electrical diagram builder	FSTRUC, FGRP, TAG
Free Drawing	DRAWING	Diagram builder	FSTRUC, FGRP, TAG
Assembly Drawing	HOOKUP	Diagram builder	FGRP, TAG
Arrangement Drawing	LA_DIA	Diagram builder	TAG
Free Text	TEXT	Document builder	FSTRUC, FGRP, TAG

Engineering of Circuit Diagrams for Tags

Circuit diagrams of tags are assigned as children to TAG-objects. Typicals of circuit diagrams should have been created and made available in a library. The following data are inserted in the used circuit diagram typicals:

- device designations (derived from the designation of the device function objects),
- constructional units designation (information extracted from the arrangement data),
- DCS-signal element designations (equal to the object id of the signal element objects in the relevant TAG),
- DCS-board designation of input and output boards (information extracted from the links from the signal element objects to the board element objects),
- designation of terminal bars and terminals (information extracted from the interconnection data).

These data are not completely available until the termination of the arrangement and interconnection engineering (see Section 6.3, Detailed Design of Arrangement and Interconnection).

The circuit diagram is assigned as child to the TAG-object by the Structure Builder and/or Template Builder, and is edited by the Electrical Diagram Builder.

Figure 6-3 shows an example of a circuit diagram as presented by the Electrical Diagram Builder.

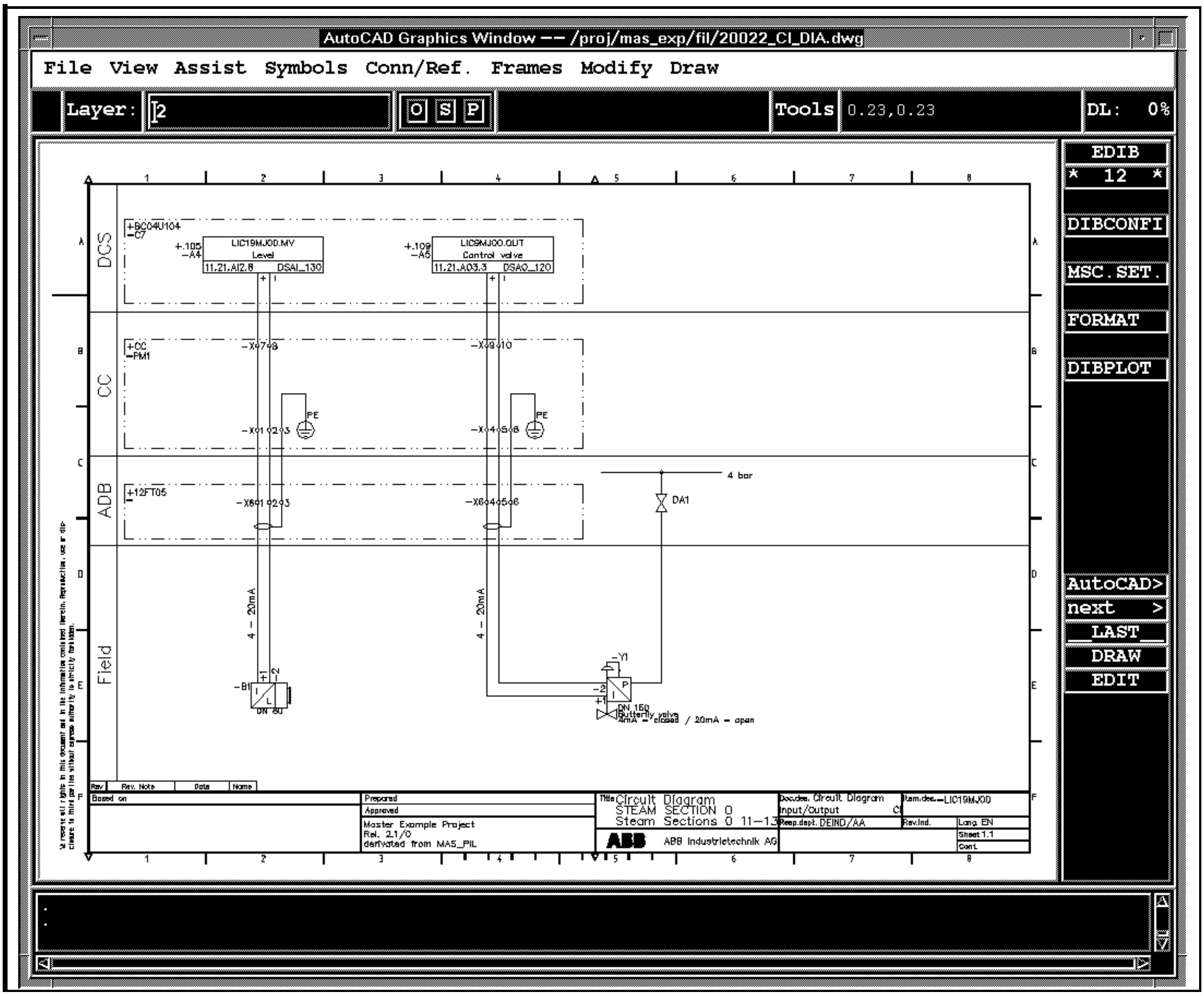


Figure 6-3. Circuit Diagram

6.2 Detailed Configuration of the Control System

Detailed tag engineering delivers the number and type of input and output signals. Based on this information, the number and type of input and output board element objects can be selected and assigned as children to the MP200 respective MA90 stations as child-objects.

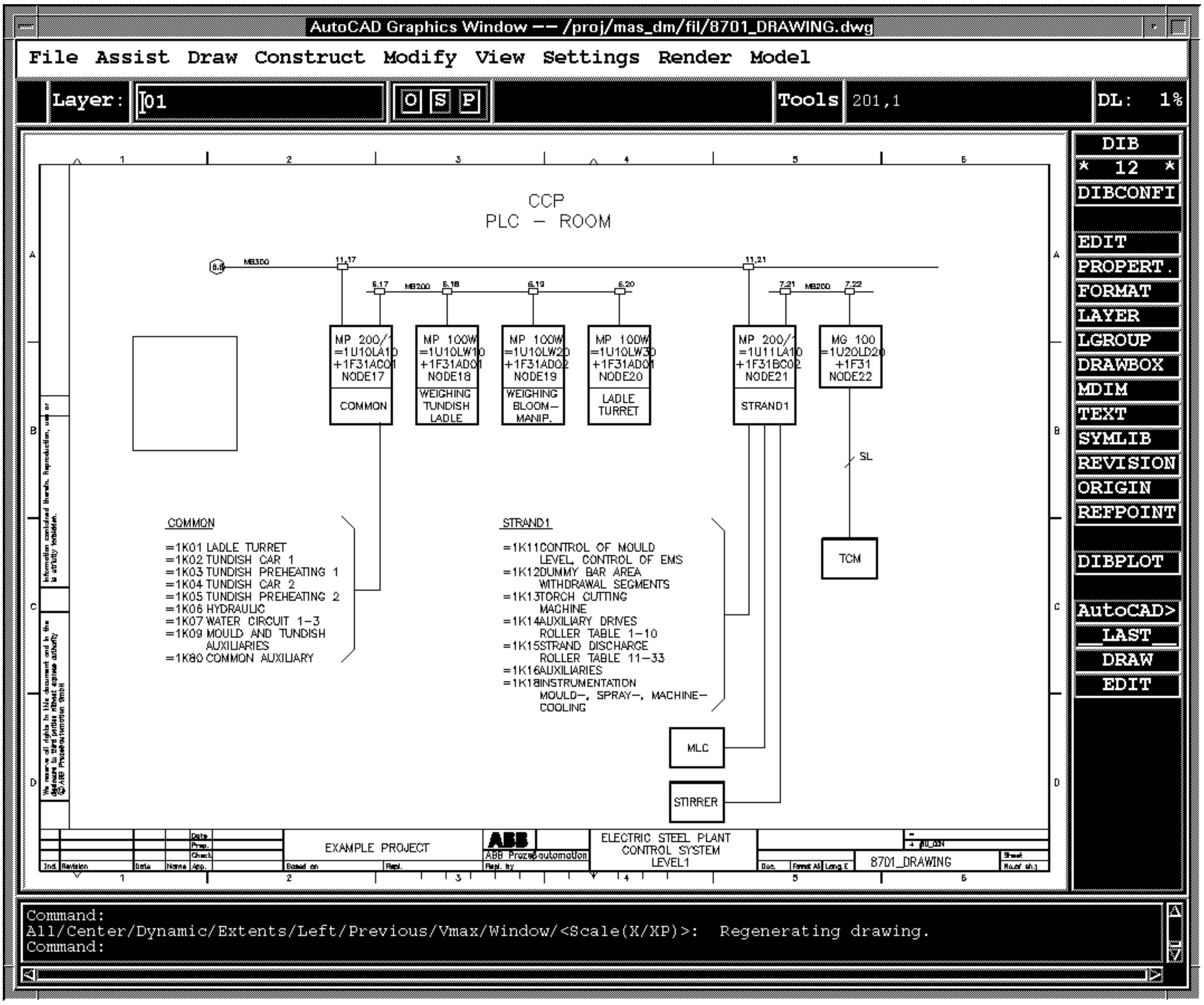
The required resources for instantiating the PC programs can be derived from the kind and number of individual control functions for tags, from the control system hierarchy, and from the preliminary specified coordination control functions.

Due to the evaluation of these data, a redesign of the preliminary control system structure may be necessary.

User activities for redesigning:

- Insertion of control system objects,
- Deletion of control system objects,
- Redesign of overview diagram "Arrangement and Interconnection of Control System".

Figure 6-4 shows an example of an Overview Diagram "Arrangement and Interconnection of Control System" as presented by the Diagram Builder.


AutoCAD Graphics Window -- /proj/mas_dm/fil/8701_DRAWING.dwg
File Assist Draw Construct Modify View Settings Render Model
Layer: 01
O S P
Tools 201,1
DL: 1%
DIB
* 12 *
DIBCONFI
EDIT
PROPERT.
FORMAT
LAYER
LGROUP
DRAWBOX
MDIM
TEXT
SYMLIB
REVISION
ORIGIN
REFPOINT
DIBPLOT
AutoCAD>
LAST
DRAW
EDIT
CCP
PLC – ROOM
MB300
11,17
11,21
6.17
MB200
6.18
6.19
6.20
7.21
MB200
7.22
MP 200/1 =1U10LA10 +1F31AC01 NODE17
COMMON
MP 100W =1U10LW10 +1F31AD01 NODE18
WEIGHING TUNDISH LADLE
MP 100W =1U10LW20 +1F31AD02 NODE19
WEIGHING BLOOM–MANIP.
MP 100W =1U10LW30 +1F31AD01 NODE20
LADLE TURRET
MP 200/1 =1U11LA10 +1F31BC02 NODE21
STRAND1
MG 100 =1U20LD20 +1F31 NODE22
SL
TCM
COMMON
=1K01 LADLE TURRET
=1K02 TUNDISH CAR 1
=1K03 TUNDISH PREHEATING 1
=1K04 TUNDISH CAR 2
=1K05 TUNDISH PREHEATING 2
=1K06 HYDRAULIC
=1K07 WATER CIRCUIT 1–3
=1K09 MOULD AND TUNDISH AUXILIARIES
=1K80 COMMON AUXILIARY
STRAND1
=1K11 CONTROL OF MOULD LEVEL, CONTROL OF EMS
=1K12 DUMMY BAR AREA WITHDRAWAL SEGMENTS
=1K13 TORCH CUTTING MACHINE
=1K14 AUXILIARY DRIVES ROLLER TABLE 1–10
=1K15 STRAND DISCHARGE ROLLER TABLE 11–33
=1K16 AUXILIARIES
=1K18 INSTRUMENTATION MOULD–, SPRAY–, MACHINE–COOLING
MLC
STIRRER
EXAMPLE PROJECT
ABB
ABB Prozessautomation
ELECTRIC STEEL PLANT
CONTROL SYSTEM
LEVEL1
8701_DRAWING
1
2
3
4
5
6
A
B
C
D

```
Command:
All/Center/Dynamic/Extents/Left/Previous/Vmax/Window/<Scale(X/XP)>:  Regenerating drawing.
Command:
```

Figure 6-4. Overview Diagram "Arrangement and Interconnection of Control System"

6.3 Detailed Design of Arrangement and Interconnection

6.3.1 Design of Constructional Units and Device Arrangement

Normally a redesign of the basic arrangement planning has to be done using more accurate information, derived from the detail planning of the peripheral equipment allocated to tags and from the detailed configuration of the control system.

The final types of constructional units (such as cubicles and desks) will be selected. The device function objects allocated to tags and the DCS-board element objects allocated to the control system are implemented in devices which have to be located in the selected constructional units. The assignment of the devices to constructional units may be accomplished stepwise. First, the device-objects are created as children of a structure-object in the mounting-location structure (it may be a dummy object in the first step, such as a dummy constructional unit) and linked to the corresponding board element objects in the control system structure.

The assignment of devices to the final mounting locations may be done in a recurrent approach. The constructional units will be subdivided into subunits and racks. This is done by creating corresponding objects using the Structure Builder or by copying standard substructures from a project library. The object id's have to be in accordance with the designation rules for mounting locations.

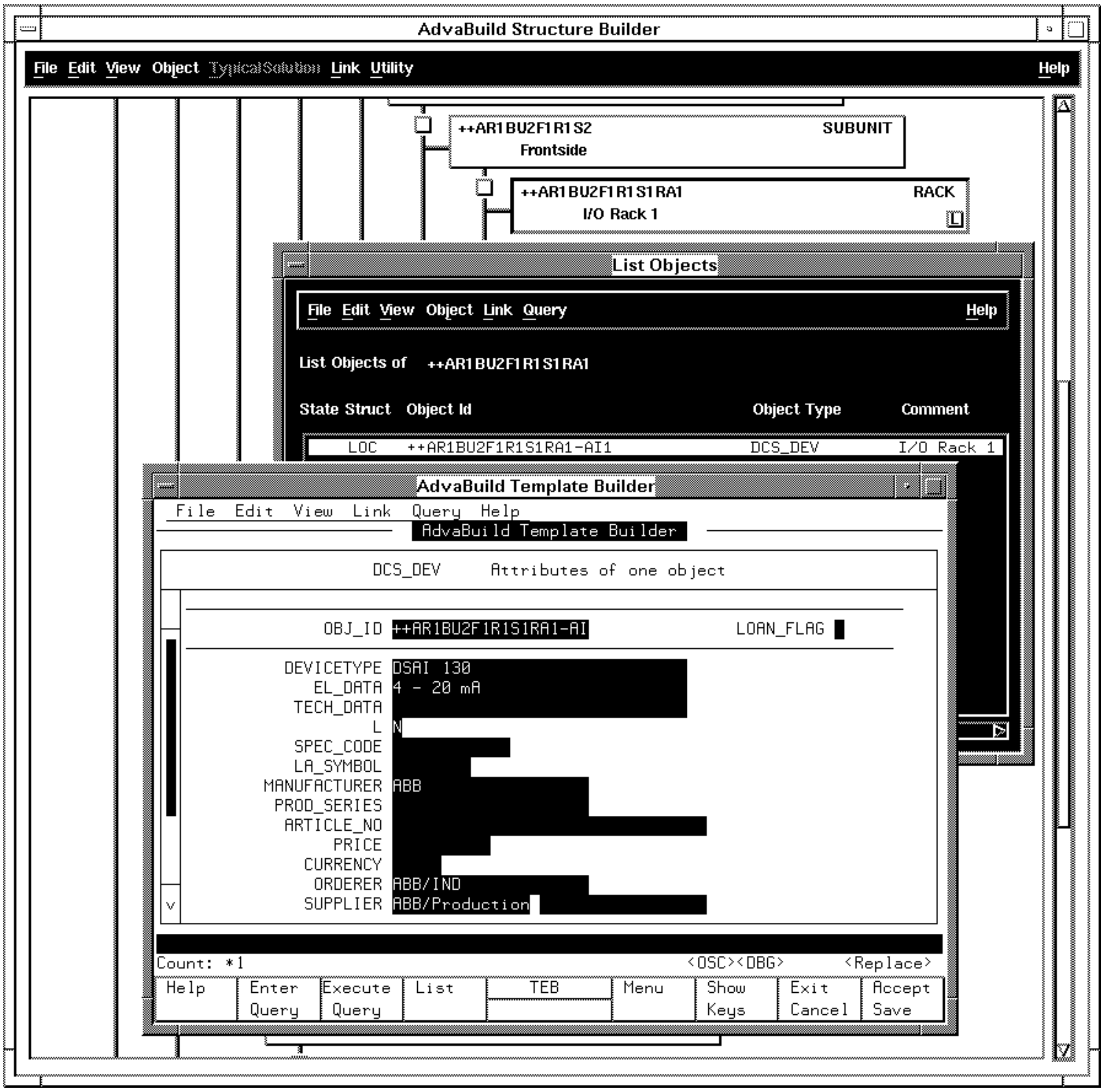
The devices can be allocated to their final destination after creating all mounting locations per constructional unit. Devices are moved from the preliminary location-object to the final location (such as a rack in the constructional unit) via the **Move** function of the Structure Builder.

Inserting a DCS-device into a mounting location (for list presentation).

User activities:

- Log in to the Structure Builder.
- Click on mounting-location object to which DCS-device shall be assigned.
- Choose **Open Type** from the **Object** menu.
- Select DCS-device from selectorbox.
- Choose **Insert** from the **Edit** menu in the Template Builder window.
- Input object designation and attribute values.
- Click on **Accept/Save.**

The presentation of a DCS-device object and its attributes by the Structure Builder and Template Builder is shown in Figure 6-5.


AdvaBuild Structure Builder
File Edit View Object TypicalSolution Link Utility Help
List Objects
File Edit View Object Link Query Help
AdvaBuild Template Builder
File Edit View Link Query Help
AdvaBuild Template Builder
Count: *1
<OSC><DBG>
<Replace>
Help
Enter Query
Execute Query
List
TEB
Menu
Show Keys
Exit Cancel
Accept Save

++AR1BU2F1R1S2 SUBUNIT
Frontside

++AR1BU2F1R1S1RA1 RACK
I/O Rack 1

List Objects of ++AR1BU2F1R1S1RA1

State	Struct	Object Id	Object Type	Comment
	LOC	++AR1BU2F1R1S1RA1-AI1	DCS_DEV	I/O Rack 1

DCS_DEV Attributes of one object

OBJ_ID ++AR1BU2F1R1S1RA1-AI
LOAN_FLAG
DEVICETYPE DSAI 130
EL_DATA 4 - 20 mA
TECH_DATA
L N
SPEC_CODE
LA_SYMBOL
MANUFACTURER ABB
PROD_SERIES
ARTICLE_NO
PRICE
CURRENCY
ORDERER ABB/IND
SUPPLIER ABB/Production

Figure 6-5. Presentation of a DCS-Device Object by Structure Builder and Template Builder

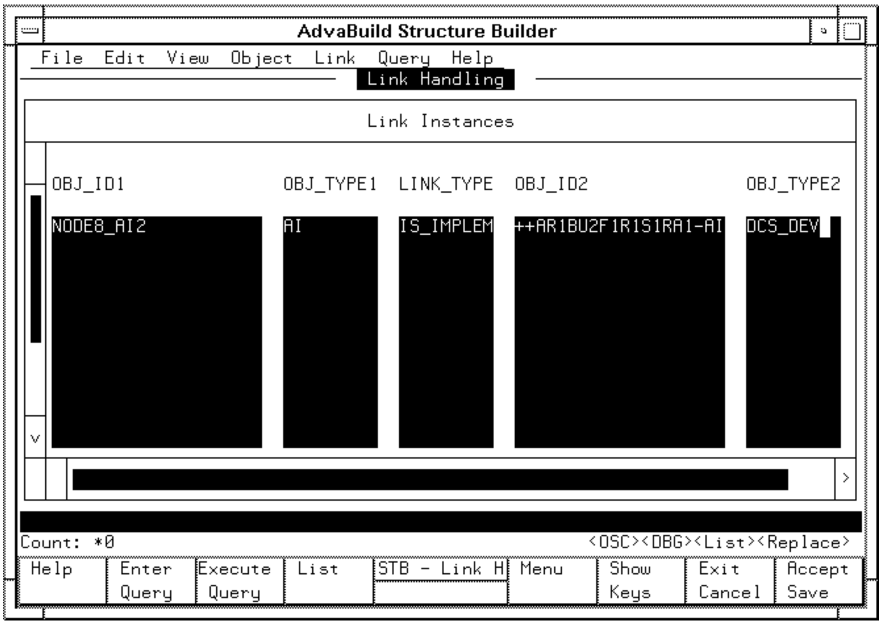
DCS-device objects are linked to board element objects in the control system structure by the link-type IS_IMPLEMENTED_BY.

Creation of a link between a DCS-board and a DCS-device.

User activities:

- Log in to the Structure Builder.
- Select board element object in control system structure.
- Select menu item **Link** and submenu item **of Object.**
- Select link-type IS_IMPLEMENTED_BY from selectorbox.
- Insert object id's of board and DCS-device in the Link Instance window.
- Close link by clicking **Accept/Save.**

The links can be created and indicated in the General Link Handling window as shown in Figure 6-6:



OBJ_ID1	OBJ_TYPE1	LINK_TYPE	OBJ_ID2	OBJ_TYPE2
NODE8_AI2	AI	IS_IMPLEM	++AR1BU2F1R1S1RA1-AI	DCS_DEV

Count: *0 <OSC><DBG><List><Replace>

Figure 6-6. General Link Handling Window

Information regarding arrangement inside constructional units can be derived from the substructure of the constructional unit and from the objects related to it. A comprehensive presentation in list form may be defined as Report (document-type: Arrangement List).

The final circuits with the complete information have to be presented in the circuit diagrams for a tag. When the arrangement engineering for devices has been completed, the final location designation for mounting-location objects and devices can be transferred to the circuit diagrams. If the object id's correspond to the rules of the applied designation system, the mounting-location object id can be used as mounting-location designation.

Results of the arrangement engineering within the mounting-location structure are presented in the following documents:

- arrangement Drawing - Items in Constructional Units
- arrangement Lists - Items in Constructional Units.

The corresponding document-objects are assigned to mounting-location objects.

6.3.2 Arrangement of Constructional Units and Components

The final assignment of constructional units (such as cubicles and desks) to topographic locations is performed by moving these objects to the selected topographic-location objects.

Moving a constructional unit to its final topographic location.

User activities:

- Log in to the Structure Builder.
- Click on mounting-location object to be moved.
- Choose **Move** from the **Edit** menu.
- Insert object id of new parent-object.
- Move object.

Results of arrangement engineering within the topographic-location structure are presented in the following documents:

- arrangement and Interconnection Drawing
- arrangement Lists
- item Location Diagram.

The documents listed above are assigned to TSTRUC-objects.

6.3.3 Detailed Interconnection Engineering

The interconnections between constructional units (such as cubicles and desks) and between constructional units and peripheral devices (such as sensors and actuators) can be designed in detail after the completion of arrangement engineering. It comprises the selection of the cable types and the definite assignment of cables to cable routes, devices, and constructional units, and the assignment of the cable cores to terminals.

The results of these engineering procedures are presented in the following documents:

- Arrangement and Interconnection Drawing
- interconnection Diagram
- cable List
- documents for Cable Installation
- terminal Diagrams.

The documents listed above will be assigned as children to the relevant objects of the topographic-location structure and the mounting-location structure. They are inserted by the Structure Builder and/or the Template Builder and edited by the Document Builder respective Diagram Builder.

6.4 Design of Mechanical Assemblies (Hook-ups)

Typical mechanical mounting assemblies for measuring- and actuating-devices or other peripheral devices are created by the Diagram Builder. Typicals should be available in a library.

The result is presented in the document-type "Assembly Drawing". The corresponding document-object is assigned to the relevant TAG-object.

User activities:

- insert HOOKUP-object by using the Structure Builder.
- edit HOOKUP-object by using the Diagram Builder (see Figure 6-7).

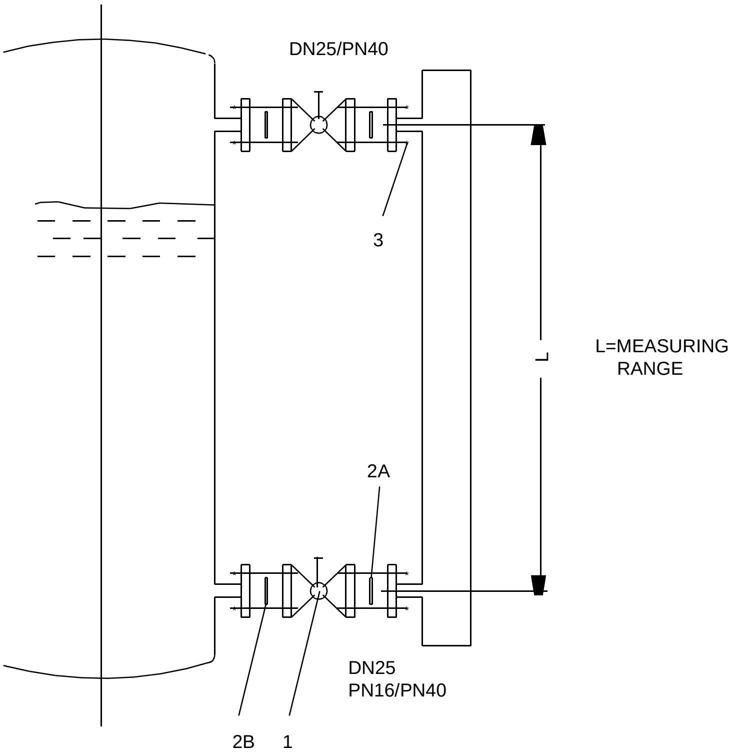


Figure 6-7. Assembly Drawing (Hook-up)

6.5 Detail Design of Control Functions

6.5.1 Handling of Type Circuits

Type Circuits are preconfigured and tested solutions for control functions and associated database elements. They are created by the configuration system MA220 and imported in AMPL source code to a predefined directory in the workstation.

Type Circuits which can be used for the project under consideration are imported to the workstation (Section 5.4.1, Type Circuits). Before configuring control functions for a node of the control system, that node has to be initialized by selecting which PC-element library shall be used and which Type Circuits out of the type circuit directory should be attached to the PC-element library.

Type Circuits will be used within function configuration and are inserted by the user like PC-elements.

Implementation

Attachment of Type Circuits to the PC-element library for a node.	
Preconditions	• Type Circuits available in TC source code in predefined directory of the workstation.
User activities:	• Use the Function Chart Builder. • Specify system version of node. • Select PC-element libraries by dialog. • Select Type Circuits by dialog. • Close the 'element library compiler' window.
Result:	• Type Circuits attached to the element library of a node.

6.5.2 Assignment of Control Functions to Control System Objects (Nodes)

The control functions are implemented as PC-programs and DB-elements and assigned as children to the control system nodes MP200 or MP90. The control functions specified in basic engineering may first have to be redesigned according to the chosen solutions in detailed engineering.

Implementation

Inserting function-objects for control functions (PCPGM-objects) as children to MP200 or MP90 nodes.

Preconditions:

- Control functions defined
- Control system structure established.

User activities:

- Use the Structure Builder.
- Open MP200 (or MP90)-object and select view Function Chart Builder.
- Choose **Function Chart Repr.** from the **Edit** menu.
- Enter item designation of PC-program (such as PC3).
- Exit Function Chart Builder.

Result:

PCPGM-object inserted.

PC-programs may also be inserted to the control system structure by using only the Structure Builder or Template Builder.

6.5.3 Configuration of Control Functions

The control functions which were defined in an overview manner in basic engineering have to be designed in all details. This may be accomplished in two steps.

First, in an intermediate step, the functions or part of the functions may be specified by the Diagram Builder using standardized symbols for function representation (according to IEC standards), and/or by the Document Builder using text. This planning step is necessary when further parties are involved in the planning work and when common means of communication were agreed upon for general standards and text.

For creation, alteration, and deletion of these function-representations, the same procedure is applied as explained for the overview specification of functions in basic engineering phase (see Section 5.5, Basic Function Planning and Acquisition of Basic Engineering-data). Note that the function-objects of basic engineering have to remain established in AdvaBuild with their own revision administration, and separate objects must be created for detailed specification of these functions.

The engineering step which handles those function-objects which are fully integrated into the AdvaBuild data representation of the plant is performed by the Function Chart Builder.

Implementation

Configuration of control functions as PC-programs.	
Preconditions:	• Control functions specified • Control system structure established • PC-programs to MP200 or MP90 assigned • DCS-signals specified.
User activities	• Use the StructureBuilder. • Open MP200 (or MP90)-object and select view Function Chart Builder. • Choose **Function Chart Repr.** from the **Edit** menu. • Select PC-program (such as PC3). • Insert PC-elements. • Connect PC-elements, DB-elements. • Exit Function Chart Builder.
Result:	• Configured PC-programs (PCPGM-objects).

Opening a node-object provides access to all tool functions of the Function Chart Builder. These tool functions are:

- PC EDITOR,
- CODE GENERATION,
- BACKTRANSLATION,
- DOCUMENT GENERATION.

Opening a PC program-object provides access to the PC EDITOR only.

When using a tool function of the Function Chart Builder, it is possible to navigate within the PC-programs of one node. It is not possible to switch to another node.

Figure 6-8 shows an example of a function chart as presented by the Function Chart Builder:

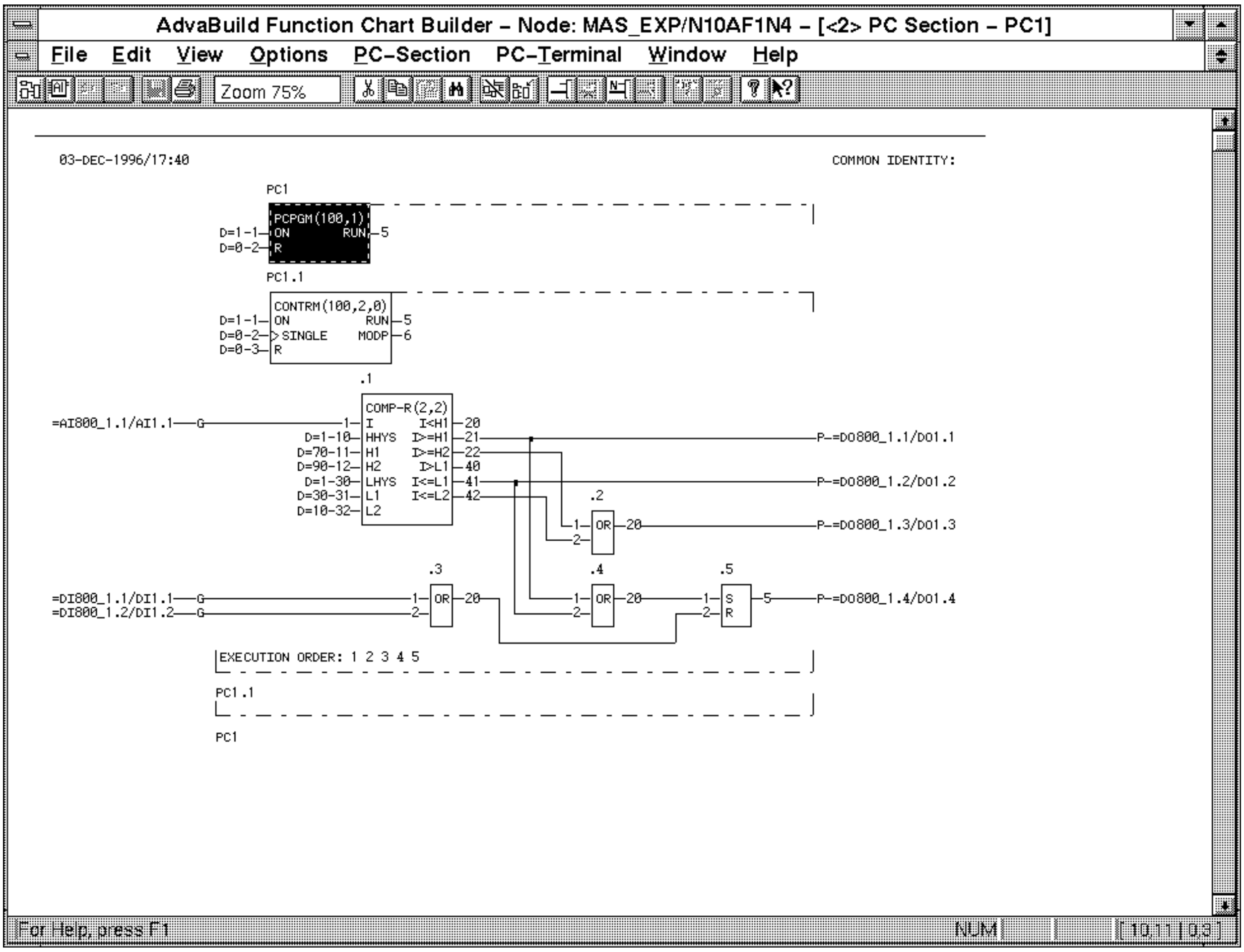


Figure 6-8. Function Chart

6.5.4 Insertion of MasterPiece Data base Elements

Signals and related data which have to be globally accessible (within one node of the control system) are stored within the control system in database elements. For input and output signals to peripheral devices, the signal database elements are linked to channels of board elements.

DCS-signal element objects may be inserted either in the function structure or in the control system structure. In the function structure, the DCS-signal element objects are placed as child-objects to the TAG-objects (see Section 6.1.1.1, DCS-signal Elements). In the control system structure, they are placed as children to the node-object MP200 or MP90.

Each DB-element type is represented by one object-type. The object type name is identical to the call name of the corresponding DB-element. All defined DB-element types are presented in a selector box during the insertion activity.

Implementation

Insertion of DB- elements to MP200 or MP90 nodes in the control system structure and insertion of their attributes (insertion procedure for list presentation).	
Preconditions:	• Tags defined • Control system structure created.
User activities:	• Use the Structure Builder. • Click on node (such as MP200-object). • Choose **Open Type** from the **Object** menu. • Select object-type to be inserted from the selector box (such as AIS). • Choose **Insert** from the **Edit** menu in the Template Builder window. • Input object designation and attribute values. • Click on **Accept/Save.**
Result:	• DCS-signal and its attributes inserted to control system node.

6.5.5 Linking a DCS-signal Object to an Open Channel of a Board-object

The relationship between a DCS-signal element object and a board-element object is described with the special link-type HAS_SIGNAL_ELEMENT. This link-type is only used for this purpose. Such links will be created automatically as open links after a board-element object has been created. A link of this type is closed by assigning a DCS-signal element to a board element channel. There are two ways to do this. The first method is used if there are only a few links to create. The second method is suited for bulk data.

Creation of a link between a signal element object and a board element object.	
Preconditions:	• Board element object and signal element object which shall be linked are created.
User activities:	• Use the Structure Builder. • Click on board-object in control system structure. • Choose **Of Object** from the **Link** menu. • Select link-type HAS_SIGNAL_ELEMENT from selectorbox. • Select board channel(s) and assign object id(s) of signal element object(s) to board channel(s) in Link Handling window. • Click on **Accept/Save.**
Result:	• Link between board element object(s) and signal element object(s) closed.

Semi-automatic creation of links between signal element objects and board element objects (for bulk data).

Preconditions:

- Board element objects and signal element objects which should be linked already created.

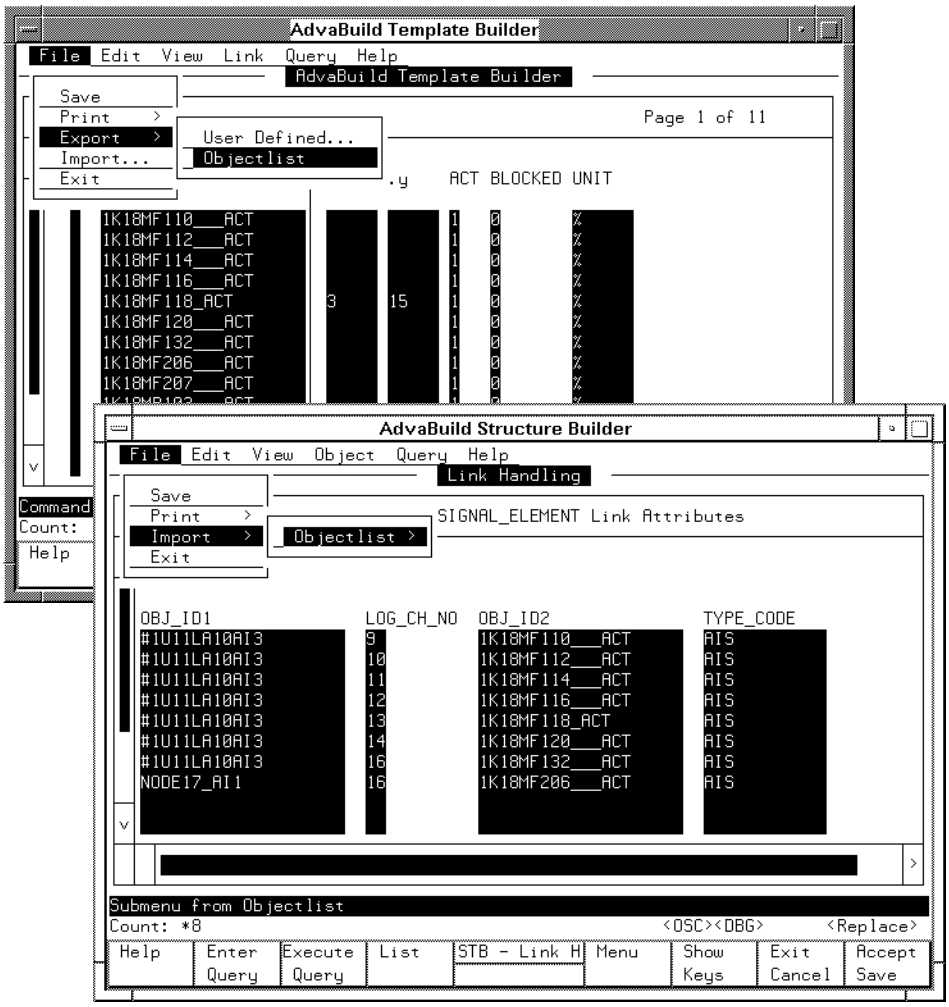
User activities:

- Use the Structure Builder.
- Click on node in the control system structure to which signal elements shall be linked.
- Choose **Of Whole Subtree** from the **Link** menu.
- Select link-type HAS_SIGNAL_ELEMENT from selectorbox.
- Choose **Set Dir.** from the **View** menu in the Link Handling window.
- Input query condition into dialog box for obj_id1, such as query for all open links of all AI boards in the selected node.
- Execute the query.
- Open the Template Builder and query for the signal objects of interest, such as AIS.
- Choose **Export > Objectlist** from the **File** menu in the TEB window.
- Choose **Import > Objectlist**> **Into Obj_id2** from the **File** menu in the Link Handling window.
- Import by clicking on **Accept/Save.**
- Insert links by clicking on **Accept/Save.**

Result:

- Links between free channels of board element objects and selected signal element objects closed.

An example for the semi-automatic linking of signal elements to board elements shows Figure 6-9.



Page 1 of 11

		.y	ACT	BLOCKED	UNIT
1K18MF110___ACT			1	0	%
1K18MF112___ACT			1	0	%
1K18MF114___ACT			1	0	%
1K18MF116___ACT			1	0	%
1K18MF118_ACT	3	15	1	0	%
1K18MF120___ACT			1	0	%
1K18MF132___ACT			1	0	%
1K18MF206___ACT			1	0	%
1K18MF207___ACT			1	0	%

SIGNAL_ELEMENT Link Attributes

OBJ_ID1	LOG_CH_NO	OBJ_ID2	TYPE_CODE
#1U11LA10AI3	9	1K18MF110___ACT	AIS
#1U11LA10AI3	10	1K18MF112___ACT	AIS
#1U11LA10AI3	11	1K18MF114___ACT	AIS
#1U11LA10AI3	12	1K18MF116___ACT	AIS
#1U11LA10AI3	13	1K18MF118_ACT	AIS
#1U11LA10AI3	14	1K18MF120___ACT	AIS
#1U11LA10AI3	16	1K18MF132___ACT	AIS
NODE17_AI1	16	1K18MF206___ACT	AIS

Submenu from Objectlist
Count: *8 <OSC><DBG> <Replace>

Figure 6-9. Link Handling for Bulk Data

6.5.6 Loading of PC-programs and DB-elements to the Target System

Engineering data which specify the PC-programs and the database for the MASTER target system have to be translated to the loadable code and loaded to the target system. This is accomplished either by using the **configuration system MA220** or the **On-Line-Builder.**

Loading the Target System via the Configuration System MA220

The loading is accomplished in three main steps. In the first step a source code is generated as input for the configuration system (MA220 for MASTER). In the second step this code is transferred to the MA220 system. In the third step the code for the target system is generated by MA220 and downloaded.

Loading PC-programs and DB-elements of a node from the workstation to the target system via the configuration system MA220.

Preconditions:

- PC-programs and DB-elements configured in the workstation
- Data transmission line between the workstation and MA220 installed and systems initiated for data transmission.

User activities:

- Open node-object and select view Function Chart Builder.
- Choose **Translate** from the Function Chart Builder main menu.
- Select **Complete Node** from the **GenerateSource** menu.
- Enter DB code generator options (such as '/D').
- Enter filename for DB sourcecodefile.
- Enter filename for PC sourcecodefile.
- Copy sourcecode files to virtual disc on workstation.
- Open project on MA220 system.
- Initialize project in function chart editor of MA220.
- Copy sourcecodefiles from virtual disc to MA220 project.
- Prepare sourcecode for MA220.
- Load files from MA220 into target system.

Result:

- Target system node loaded.

Loading the Target System via the On-Line-Builder

Loading PC-programs and DB elements of a node from the workstation to the target system MP220.

Preconditions:

- PC-programs and DB-elements configured in the workstation
- Hardware communication board PU510 and data transmission line between workstation and Master Bus300 installed and initiated.

User activities:

- Open node-object and select view Function Chart Builder.
- Choose **Translate** from the Function Chart Builder main menu.
- Select **Complete Node** from the **GenerateSource** menu.
- Enter DB code generator options (such as ‘/D’).
- Enter filename for DB sourcecodefile.
- Enter filename for PC sourcecodefile.
- Open node-object and select view On-Line-Builder (TARGET CTRL).
- Select PCDATA files and DBDATA files to be prepared for loading.
- Click on OK.
- Load PCDATA files and DBDATA files via the On-Line- Builder Target Control window to the target node.

Result:

- Target system node loaded.

6.6 Creating Displays

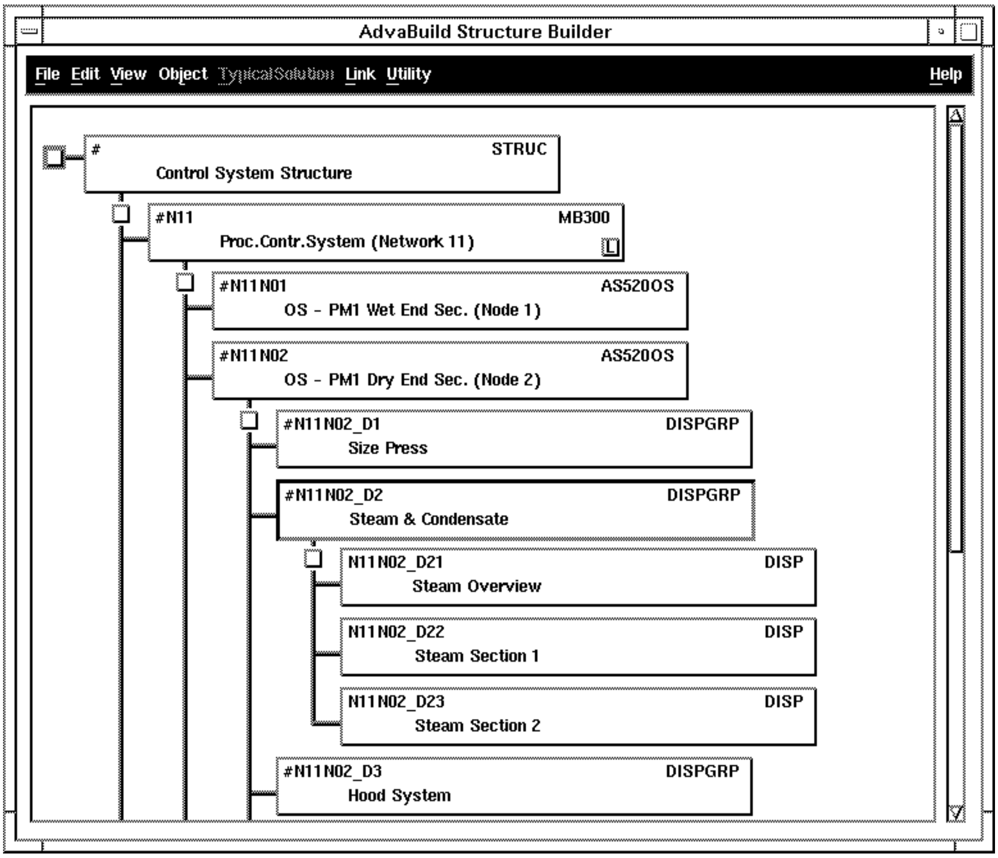
Process displays on the operator station are the interface between the operator and the process. These displays can be created and edited on the engineering station and are downloaded to the operator station.

Process displays represent the technical realization of a process by means of graphical symbols for equipment and piping, together with graphical symbols for process measurement and control functions. Displays have on-line connection to process variables for visualization and operation on the defined dynamic parts of the display when implemented on the operator station.

Basic input information for the creation of displays are process flow diagrams and pipe and instrumentation (P&I) diagrams.

Displays are represented by DISP-objects and are inserted as children objects to DISPGRP-objects, which represent a display group. DISPGRP-objects are children of AS500OIS-objects.

Figure 6-10 shows an example of the assignment of DISP-objects in the control system structure.


AdvaBuild Structure Builder
File Edit View Object TypicalSolution Link Utility Help

- # STRUC Control System Structure
  - #N11 MB300 Proc.Contr.System (Network 11)
    - #N11N01 AS520OS OS - PM1 Wet End Sec. (Node 1)
    - #N11N02 AS520OS OS - PM1 Dry End Sec. (Node 2)
      - #N11N02_D1 DISPGRP Size Press
      - #N11N02_D2 DISPGRP Steam & Condensate
        - N11N02_D21 DISP Steam Overview
        - N11N02_D22 DISP Steam Section 1
        - N11N02_D23 DISP Steam Section 2
      - #N11N02_D3 DISPGRP Hood System

Figure 6-10. Display Objects in the Control System Structure

Inserting and editing a DISP-object.

Preconditions:

- DISPGRP-object created, to which DISPLAY-object can be assigned.

User activities:

- Use the Structure Builder.
- Click on the DISPGRP object to which the DISP-object shall be assigned.
- Choose **Insert** from the **Edit** menu.
- Select object-type DISPLAY, designate it uniquely, and comment it.
- Insert DISPLAY-object.
- Open DISP-object, select view DISPLAY.
- Select template, such as PROCESSdisplay.
- Edit and save display.
- Exit Display Builder.

Result:

- DISPLAY-object inserted and edited.

Figure 6-11 shows a Process Display as presented by the Display Builder.

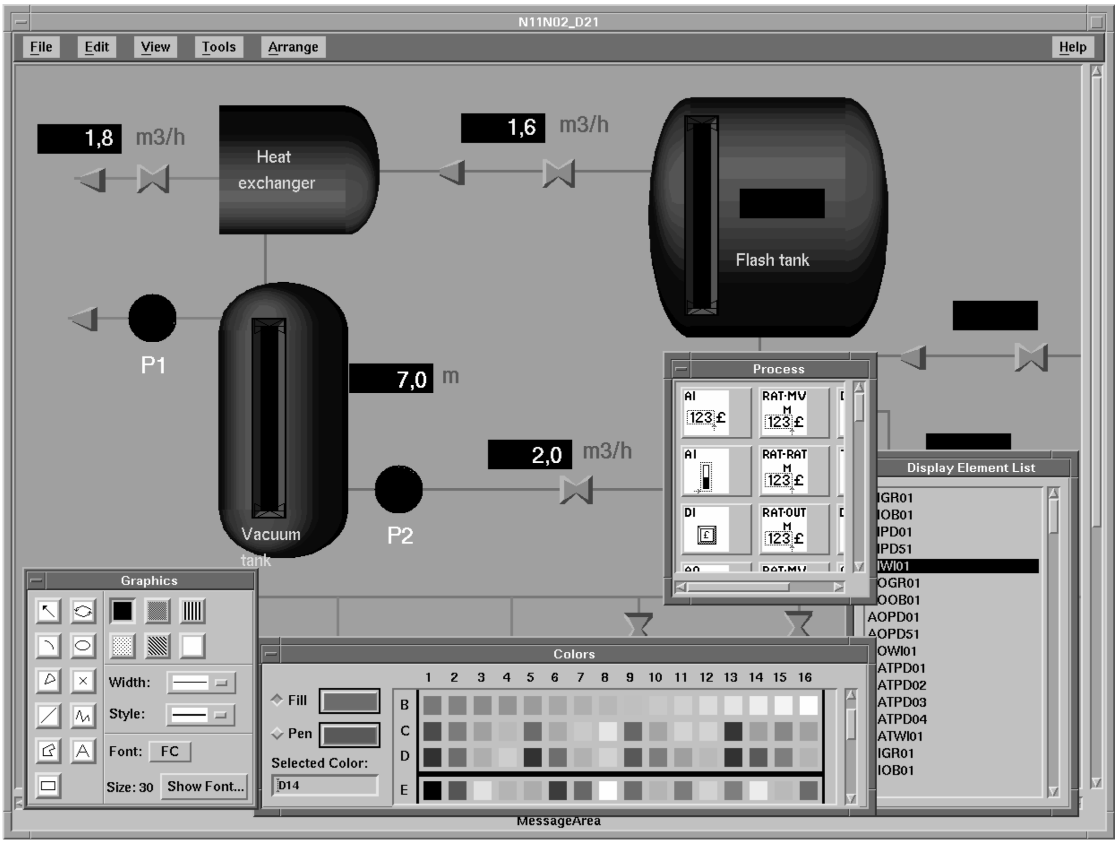


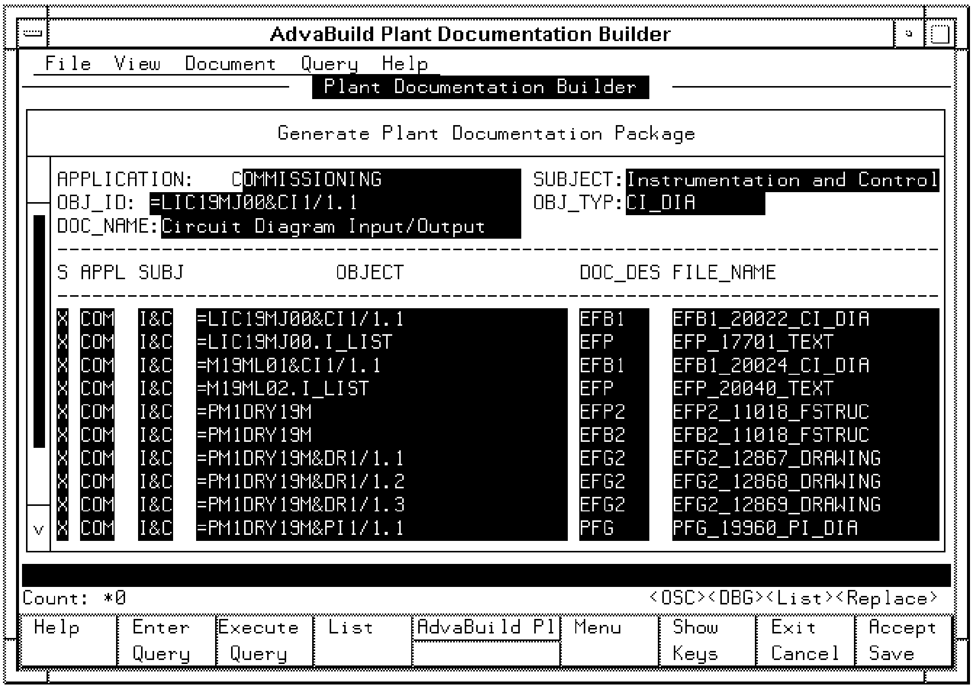
Figure 6-11. Process Display

6.7 Plant Documentation

Plant documentation packages are defined during plant documentation planning (see Section 5.8, Planning of Documentation) and are linked to the function structure and/or control system structure and/or location structure. The links of a plant documentation package define which parts of the delivery are covered by the plant documentation package.

Output of a plant document package (object of the documentation structure).	
Preconditions:	• Plant documentation structure created • Plant documentation package linked.
User activities:	• Use the Structure Builder. • Click on plant document package. • Choose **Plant Documentation Package** from the **Utility** menu. (The Generation window of the Plant Documentation Builder appears and displays the list of documents defined for the selected package.) • Select/deselect document-types within the predefined package. – Move cursor into select flag column of displayed document list. – Replace 'x' by ' ' for those documents which shall not be printed, or – Deselect all documents by choosing **Deselect All** from the **Document** menu, and replace ' ' with 'x' for those documents which shall be printed. – Click on **Accept/Save.** • Choose **Print** from the **Document** menu.
Result:	• Documents defined for the selected package and print output for the subtree(s) to which the plant document package is linked.

The documents that are selected for printout are displayed in the Package Generation Window of the Plant Documentation Builder, Figure 6-12.


AdvaBuild Plant Documentation Builder
File View Document Query Help
Plant Documentation Builder
Generate Plant Documentation Package
Count: *0
<OSC><DBG><List><Replace>
Help
Enter Query
Execute Query
List
AdvaBuild Pl
Menu
Show Keys
Exit Cancel
Accept Save

APPLICATION: COMMISSIONING
SUBJECT: Instrumentation and Control
OBJ_ID: =LIC19MJ00&CI1/1.1
OBJ_TYP: CI_DIA
DOC_NAME: Circuit Diagram Input/Output

S	APPL	SUBJ	OBJECT	DOC_DES	FILE_NAME
X	COM	I&C	=LIC19MJ00&CI1/1.1	EFB1	EFB1_20022_CI_DIA
X	COM	I&C	=LIC19MJ00.I_LIST	EFP	EFP_17701_TEXT
X	COM	I&C	=M19ML01&CI1/1.1	EFB1	EFB1_20024_CI_DIA
X	COM	I&C	=M19ML02.I_LIST	EFP	EFP_20040_TEXT
X	COM	I&C	=PM1DRY19M	EFP2	EFP2_11018_FSTRUC
X	COM	I&C	=PM1DRY19M	EFB2	EFB2_11018_FSTRUC
X	COM	I&C	=PM1DRY19M&DR1/1.1	EFG2	EFG2_12867_DRAWING
X	COM	I&C	=PM1DRY19M&DR1/1.2	EFG2	EFG2_12868_DRAWING
X	COM	I&C	=PM1DRY19M&DR1/1.3	EFG2	EFG2_12869_DRAWING
X	COM	I&C	=PM1DRY19M&PI1/1.1	PFG	PFG_19960_PI_DIA

Figure 6-12. Document Package Generation Window

There are additional ways to print out documentation. For more information see the user's manual for the Plant Documentation Builder.

Appendix A Overview of Object-types and Link-types

Table A-1. AdvaBuild Object Types

CLASSIFICATION	**OBJECT-TYPE**	**Assignment to Structure-types**					
		=	**++/+**	**#**	**&**	**Lib**	**Sol**
Object Types for Basic Project Structuring	PROJECT						
	STRUC	X	X	X	X	X	X
	DSTRUC				X		
	FSTRUC	X					
	LSTRUC					X	
	TSTRUC		X				
	MSTRUC		X				
Object Types representing Plant Functions	FSTRUC/	X					
	FGRP/	X					
	TAG/	X					
	DEV_FUN/	X				X	
Object Types representing Articles	CUNIT		X			X	
	SUBUNIT		X			X	
	RACK		X			X	
	DEV	X	X			X	
	DCS_DEV		X			X	
	BAR		X			X	
	CABLE	X	X			X	
	ACCESSORY	X	X	X			
Object Type representing Documentation	DSTRUC				x		
Object Types representing CAD Documents	HOOKUP	X				X	
	DRAWING	X	X	X		X	
	CI_DIA	X	X			X	
	PI_DIA	X				X	
	LA_DIA		X			X	

Table A-1. AdvaBuild Object Types (Continued)

CLASSIFICATION	**OBJECT-TYPE**	**Assignment to Structure-types**					
		=	**++/+**	**#**	**&**	**Lib**	**Sol**
Object Type representing Text Documents	TEXT	X	X	X		X	
Library Objects	LSTRUC		X			X	
	CUNIT		X			X	
	SUBUNIT		X			X	
	RACK		X			X	
	BAR		X			X	
	DEV	X	X			X	
	DCS_DEV		X			X	
	TERMINAL					X	
	CHANNEL					X	
	CPOINT					X	
	CABLE	X	X			X	
	CORE					X	

Table A-1. AdvaBuild Object Types (Continued)

CLASSIFICATION	**OBJECT-TYPE**	**Assignment to Structure-types**					
		=	**++/+**	**#**	**&**	**Lib**	**Sol**
Object Types for MP200, AC410, and AC450 DB Elements	Node/Station DB Element Object			X			
	Processor Module Element Objects			X			
	Submodule Element Objects			X			
	Defining/Controlling Elements			X			
	Local General I/O Board Element Objects			X			
	Local Temperature I/O Board Element Objects			X			
	Pulse Counting I/O Board Element Objects			X			
	Distributed I/O Board Element Objects			X			
	Redundant I/O Board Element Objects			X			
	S800 I/O Module Element Objects			X			
	Communication Element Objects			X			
	Event Handling Objects			X			
	Application Oriented Objects			X			
	Function Unit Group 1 Objects			X			
	Function Unit Group 2 Objects			X			
	Function Unit Group 3 Objects			X			
	Signal Element Objects	X		X			
	BU Specific Objects			X			

Table A-1. AdvaBuild Object Types (Continued)

CLASSIFICATION	**OBJECT-TYPE**	**Assignment to Structure-types**					
		=	**++/+**	**#**	**&**	**Lib**	**Sol**
Object Types for AC110 DB Elements	Processor Module Element Objects			X			
	Module Element Objects			X			
	Submodule Element Objects			X			
	Process I/O Board Element Objects			X			
	Signal Element Objects	X		X			
	Communication Element Objects			X			
	Event Handling Objects			X			
	Miscellaneous DB Element Objects			X			
Object Types for AC70 DB Elements	Processor Module Element Objects			X			
	Process I/O Board Element Objects			X			
	Signal Element Objects	X		X			
	Communication Element Objects			X			
	Event Handling Objects			X			
	Miscellaneous DB Element Objects			X			

Table A-1. AdvaBuild Object Types (Continued)

CLASSIFICATION	**OBJECT-TYPE**	**Assignment to Structure-types**					
		=	**++/+**	**#**	**&**	**Lib**	**Sol**
Object Types representing Networks, Nets and Buses	PNET			X			
	CNET			X			
	LCNET			X			
	TCP_IP			X			
	EXT_NET			X			
	GCOM			X			
	MB300			X			
	MB200			X			
	MVI_NET			X			
	RCOM			X			
	MB90			X			
	AF100			X			
	AF100IOS			X			
	AF100SC			X			
	MFB			X			
	EXCOM			X			

Table A-1. AdvaBuild Object Types (Continued)

CLASSIFICATION	**OBJECT-TYPE**	**Assignment to Structure-types**					
		=	**++/+**	**#**	**&**	**Lib**	**Sol**
Object Types representing Nodes within Master/Advant Networks	MODGATE			X			
	MG230			X			
	MV810			X			
	MV830			X			
	MV850			X			
	SVX00			X			
	MBC200			X			
	MP200			X			
	MP51			X			
	MP40			X			
	AC55C			X			
	AC110C			X			
	AC70C			X			
	AC410C			X			
	AC450C			X			
	AS100ES			X			
	AS100MS			X			
	AS510OS			X			
	AS515OS			X			
	AS520OS			X			
	AS515IMS			X			
	AS520IMS			X			
	AS800IMS			X			
	AS515ES			X			
	AS520ES			X			
	EXT_COMPUTER			X			
Object Types for Solutions	SOLUTIONPOOL						X
	SOLUTION						X
Object Type for Special Purpose	FREE	X	X	X			

Table A-2. AdvaBuild Link-types

Link-type	Connects	With
HAS_SIGNAL_ELEMENT	Channels of I/O board objects	Signal element objects
HAS_CONNECTION_TO	Signal element objects	PCPGM
IS_DOCUMENTATION_OF	DSTRUC	Objects in the function structure, location structure, and control system structure
IS_IMPLEMENTED_BY	Board objects	DCS_DEV
HAS_LOCATION	CUNIT	SUBUNIT RACK BAR DEV DCS_DEV
	SUBUNIT	RACK BAR DEV DCS_DEV
	RACK MSTRUC	BAR DEV DCS_DEV
	(DCS_)DEV	SUBMODULE
REFERENCES_SOLUTION	All object types except SOLUTION and library objects	SOLUTION

Appendix B Target Systems Supported by AdvaBuild

Supported target systems are:

- MP200/1 V4.0
- AC 70C V1.0
- AC 110C V2.0
- AC 410C V1.2
- AC 450C V1.2.

INDEX

A
Access rights 5-1
Application of documentation 3-6, 5-20
Arrangement and Interconnection 5-18
  Control System 6-8
Arrangement and Interconnection Drawing 6-12 to 6-13
Arrangement Drawing 4-8, 4-12, 6-6
Arrangement List 6-12
Arrangement planning 5-18
Assembly Drawing 4-3, 6-6, 6-14

B
Backtranslation 6-18
BAR-object 4-12
Basic Engineering 3-1
Board element 6-7

C
Cable List 6-13
Cable Routing Drawing/List. 5-18
CABLE-object 4-12
CAPE-object 3-2
CI_DIA-object 4-3, 4-8, 4-12, 4-25, 6-6
Circuit Diagram 4-3, 4-8, 4-12, 6-6
Circuit Diagram Typical 6-6
Code Generation 6-18
Constructional unit 4-11, 5-18, 6-9
Control function 5-14, 6-16
Control function hierarchy 5-13
Control structure 5-14
Control system node 6-16
Control system structure 3-5, 4-15, 5-16
Coordination function 5-14
CUNIT-object 4-8, 4-12, 5-17

D
Danger areas 5-9
Data model 3-2
DB-element 6-16, 6-19
DCS Hardware Standards 5-4
DCS Software Standards 5-4
DCS-board element 6-9
DCS-signal element 6-1, 6-19
Design standards 5-5
Designation of objects 4-5, 4-9, 4-14
Designation prefix "#" 4-19
Designation prefix "+" (plus) 4-10
Designation prefix "=" (equal) 4-5
Designation prefix "-" (minus) 4-6
Detail Engineering 3-1
DEV_FUN-object 4-2
Device 6-3
Device function 6-3
DEV-object 4-2
Diagram Builder 2-2, 5-6, 5-14, 5-18
DIN 40719 T2 4-5, 4-9, 4-14
DIN 6779 (draft) 4-5, 4-9, 4-14
DISPGRP-object 4-16, 6-25
Display Builder 2-2
DISP-object 4-16, 6-25
Document Builder 2-2, 5-14, 5-18
Document classes 5-20
Document Generation 6-18
Document objects 6-6
Documentation package 5-23
Documentation structure 5-23
Documents for Cable Installation 6-13
DRAWING-object 4-3, 4-8, 4-12, 4-16, 4-25, 6-6
DSTRUC-object 4-4, 4-22, 5-23

E
Electrical Diagram Builder 2-2
Engineering activities 3-1

F
FGRP-object 4-2
Free Drawing 4-3, 4-8, 4-12, 4-16, 6-6
Free Text 4-3, 4-8, 4-12, 4-16, 6-6
FSTRUC-object 4-2, 5-3
Function 3-3
Function Chart Builder 2-2, 6-17
Function Description 6-1
Function Diagram 6-1
Function structure 5-3

H
HAS_LOCATION-link 4-13
HAS_SIGNAL_ELEMENT-link 4-4, 4-19, 6-1, 6-20
Higher level designation 4-5
HOOKUP-object 4-3, 4-25, 6-6, 6-14

I

IEC 750 4-5, 4-9, 4-14
Import
  P&I-Diagram 4-28
  TEXT-object 4-27
  Type Circuits 5-4
Import/export standards 5-5
Installation standards 5-5
Instrumentation List 6-1
Interconnection Diagram 6-13
Interconnections 5-18
IS_DOCUMENTATION_OF-link 4-4, 4-9, 4-13, 4-19, 4-22
IS_IMPLEMENTED_BY-link 4-13, 4-19
ISO standard 4-6
Item designation 5-4
Item Location Diagram 6-12

L

LA_DIA-object 4-8, 4-12, 4-25, 6-6
Level 4-5
Link 3-7, 4-13
  to DCS-board 6-11, 6-20
  to DCS-device 6-11
  to plant documentation package 5-26
Link-type 4-4, 4-19, 4-22
List of designations 5-2
List of general data 5-2
List of Measuring Points 6-1
List of measuring points 5-9
List presentation 4-23
Location oriented structure 3-4

M

MA220 6-22
Man machine communication 5-9
MASTER target system 6-22
MB300-object 4-16
Mechanical mounting assemblies 6-14
Motor and Component List 5-9
Mounting-location structure 3-4, 4-11, 5-17, 6-9
Move
  constructional unit 6-12
  mounting-location structure tree 5-17
MP200-object 4-16
MSTRUC-object 4-8, 4-12, 5-17
Multi-user features 4-33
MV850-object 4-16

O

Object identity 4-5
On-Line-Builder 2-2, 6-22, 6-24

P

P&I-Diagram 4-3, 5-3, 5-6, 5-9, 6-1
PC Editor 6-18
PC program 6-7
PC-element library 6-15
PCPGM-object 4-16
PC-program 6-16, 6-22
PI_DIA-object 4-3, 4-25, 5-6
PIDCON-object 4-16
Plant documentation 4-21
Plant Documentation Builder 2-2, 5-20
  Generation window 6-29
Plant documentation package 6-28
Plant documentation structure 3-6
Plant/function structure 3-3, 4-1
Process display 6-25
Process flow diagram 5-6
Project execution 3-1
Project model 3-1
Project root 5-1
Project Standards 5-3
Project-database 2-1
PROJECT-object 4-29

R

RACK-object 4-8, 4-12, 5-17
REFERENCES_SOLUTION-link 4-4, 4-9, 4-13, 4-19

S

Signal designation 4-6
Solution 3-8, 4-29
Solution Aspect Object 4-29
SOLUTIONPOOL-object 4-29
Specification Sheet Measurement and Instrumentation 5-9, 6-1
Standard documentation 5-20
STRUC-object 5-10
Structure Builder 2-1, 4-1, 4-7, 4-15, 4-21, 5-1, 5-23
Structure path 4-3, 4-6
Subfunction 3-3
Subjects of delivery 3-1, 3-6, 5-20
SUBUNIT-object 4-8, 4-12, 5-17

T
Tag function 6-1
TAG-object 4-2, 5-3, 5-6, 5-10
Template Builder 2-1, 5-1
Terminal Diagram 6-13
Test procedures 5-5
TEXT-object 4-3, 4-8, 4-12, 4-16, 4-25 to 4-27, 5-10, 6-6
Topographic view 4-7
Topographical layout 3-4
Topographic-location structure 4-7, 5-17
Transition 4-6
Tree Presentation 4-23
Tree-like structure 3-2, 4-7
TSTRUC-object 4-8, 5-17
Type circuits 5-3, 6-15

U
User-interface 2-1

V
View 3-2

W
Workstation 2-1

3BSE 001 977R0201

March, 1997